Preprint: James Mayfield, Paul McNamee, Craig Harman, Tim Finin and Dawn Lawrie, KELVIN: Extracting Knowledge from Large Text Collections, AAAI Fall Symposium on Natural Language Access to Big Data, 2014.

KELVIN: Extracting Knowledge from Large Text Collections

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

Tim Finin
Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD

Dawn Lawrie
Computer Science Department
Loyola University Maryland
Baltimore, MD

Abstract

We describe the KELVIN system for extracting entities and relations from large text collections and its use in the TAC Knowledge Base Population Cold Start task run by the U.S. National Institute of Standards and Technology. The Cold Start task starts with an empty knowledge base defined by an ontology of entity types, properties and relations. Evaluations in 2012 and 2013 were done using a collection of text from local Web and news to de-emphasize the use of entities that appear in a background knowledge base such as Wikipedia. Interesting features of KELVIN include a cross-document entity coreference module based on entity mentions, removal of suspect intra-document conference chains, a slot value consolidator for entities, the application of inference rules to expand the number of asserted facts and a set of analysis and browsing tools supporting development.

Introduction

Much information about the world is encoded in the form of text in journal articles, newswire stories, press releases, Web pages, social media posts, advertisements and email correspondence. Computer language understanding systems are able to extract entities and relations between them from such text with increasing accuracy. Once extracted, this knowledge can be added to and integrated with existing data to enhance many "big data" applications. Knowledge extracted from text also supports the creation and maintenance of the background knowledge bases needed by natural language question answering interfaces to datasets.

Since the early 1990s there has been a gradual progression in text mining research through tasks such as identifying named entities, extracting relations, linking entities to existing knowledge bases, detecting entity, relation, and event coreference, and others. The goal of these activities has been to increase the amount of structured knowledge that can be automatically extracted from naturally occurring text. However, the ability of a system to use these technologies to actually construct a knowledge base (KB) from the information provided in a text collection was not being exercised.

NIST developed TAC Cold Start Knowledge Base Population task as a way to evaluate KBs of facts extracted from large collections of text. The task's name conveys two features of the task: it implies both that a KB schema has been established at the start of the task and that the KB is initially unpopulated. Thus, we assume that a schema exists for the facts and relations that will compose the KB; it is not part of the task to automatically identify and name the types of facts and relationships present in the text collection. The task represents more than merely the combination of extant capabilities (such as slot filling and entity linking) for several reasons:

- It focuses research on errors produced by those components most important in constructing KBs from text.
- It requires systems to process large collections, facilitating research on scaling and also into joint entity resolution and coordination of extraction across slots.
- It pushes systems to reduce reliance on Wikipedia and other background KBs, which, while useful for processing English newswire articles, may not have significant coverage of other target text genres or languages.
- It facilitates research in inference over extracted knowledge and confidence estimation, which is not feasible using merely the output of individual component systems.

The JHU Human Language Technology Center of Excellence has participated in the TAC Knowledge Base Population exercise since its inception in 2009 and submitted entries for the Cold Start task in 2012 and 2013. In the remainder of this paper we briefly describe the Cold Start task and its evaluation methodology, the KELVIN (McNamee et al. 2012; 2013) system we used in Cold Start, our results on the TAC-KBP tasks, and some of the development tools we have found useful.

Cold Start KBP task

We originally developed KELVIN to participate in the 2012 Cold Start task. While many of its features were driven by this, we have been enhancing it for use on other information extraction problems. In this section we will briefly introduce the Cold Start task and how it evaluated KBs produced by participating systems.

The Cold Start task provides a KB schema and requires participants to process a large document collection (about 26,000 in 2012 and 50,000 in 2013) to extract entities and facts to populate an initially empty KB. Participants submit the KB as an unordered set of subject-predicate-object

Relation	Inverse(s)
per:children	per:parents
per:other_family	per:other_family
per:parents	per:children
per:siblings	per:siblings
per:spouse	per:spouse
per:employee_or_member_of	{org,gpe}:employees_or_members
per:schools_attended	org:students
per:city_of_birth	gpe:births_in_city
per:stateorprovince_of_birth	gpe:births_in_stateorprovince
per:country_of_birth	gpe:births_in_country
per:cities_of_residence	gpe:residents_of_city
per:statesorprovinces_of_residence	gpe:residents_of_stateorprovince
per:countries_of_residence	gpe:residents_of_country
per:city_of_death	gpe:deaths_in_city
per:stateorprovince_of_death	gpe:deaths_in_stateorprovince
per:country_of_death	gpe:deaths_in_country
org:shareholders	{per,org,gpe}:holds_shares_in
org:founded_by	{per,org,gpe}:organizations_founded
org:top_members_employees	per:top_member_employee_of
{org,gpe}:member_of	org:members
org:members	{org,gpe}:member_of
org:parents	{org,gpe}:subsidiaries
org:subsidiaries	org:parents
org:city_of_headquarters	gpe:headquarters_in_city
org:stateorprovince_of_headquarters	gpe:headquarters_in_stateorprovince
org:country_of_headquarters	gpe:headquarters_in_country

Figure 1: The 2013 Cold Start ontology comprises 26 entity-valued predicates (and their inverses) and 15 additional slots whose values are strings.

triples with associated provenance and certainty metadata. The provenance links entities and their relations to specific strings in documents.

The schema for Cold Start 2013 was derived from the TAC-KBP Slot Filling task specification (Text Analysis Conference 2013) and includes forty-one slots that cover basic biographical information about persons (*e.g.,* family and professional relationships, background), and salient properties about organizations (*e.g.,* key employees, location of facilities). Twenty-six slots have fills that are themselves entities, as shown in Figure 1; the remaining fifteen have string fills. For people these string valued slots include alternate_names, date_of_birth, age, origin, date_of_death, cause_of_death, title, religion and charges. For organizations they are alternate_names, political_religious_affiliation, number_of_employees_members, date_founded, date_dissolved, and website.

Evaluating the performance of systems that create KBs from large document collections is challenging (Mayfield and Finin 2012). Constructing a gold-standard reference KB for comparison can be expensive and, even if one is available, it must be aligned with the KB under evaluation, potentially an NP-hard task. Since Cold Start requires that all document entity mentions be tied to a KB entity node, a KB can be queried without first aligning it to the reference KB.

Evaluation uses a set of *evaluation queries*, where each query starts with a document entity mention, identifies its KB entity, and follows a sequence of one or more relations within the KB, ending in a slot fill. The resulting slot fills are assessed and scored as in traditional question answering or information extraction tasks. For example, a KB evaluation query might be *"what are the ages of the siblings of the 'Bart Simpson' mentioned in Document 42?"* A system that correctly identifies descriptions of Bart's siblings in the document collection, links them to the appropriate nodes in the KB, and finds evidence for and correctly represents their ages receives full credit. The evaluations performed by NIST consisted of a set of queries involving paths through the graph of length one or two (i.e., one or two "hops").

KELVIN

TAC KBP Cold Start is a complex task that requires application of multiple layers of NLP software. We divide these layers into three categories: document level, cross-document co-reference and inference. KELVIN runs from two Unix shell scripts that execute a pipeline of operations. The input to the system is a file listing the source documents to be processed; the files are presumed to be plain UTF-8 encoded text, possibly containing light SGML markup. During processing, the system produces a series of tab-separated files, which capture the intermediate state of the growing knowledge base. At the end of the pipeline the resulting file is compliant with the TAC KBP Cold Start guidelines.

We give examples drawn from applying KELVIN to a collection of about 26,000 newswire articles published in the Washington Post in 2010. In processing these, we followed the Cold Start task's design assumption that the domain would not be focused on well-known entities, and so we did not use an extant KB such as DBpedia or Freebase to inform processing.

Document level processing

The most significant tool that we use at the document layer is SERIF (Boschee, Weischedel, and Zamanian 2005), an ACE entity/relation/event detection system developed by BBN. This is extended with a maximum entropy trained model for extracting personal attributes (FACETS, also a BBN tool) and a system for detecting and deleting low-qualify mention chains.

Entities, mentions and relations. SERIF provides a considerable suite of document annotations that are an excellent basis for building a knowledge base. The functions SERIF can provide are based largely on the NIST ACE specification (National Institute of Standards and Technology 2008) and include: identifying named-entities and classifying them by type and subtype; performing intra-document coreference analysis, including named mentions, as well as coreferential nominal and pronominal mentions; parsing sentences and extracting intra-sentential relations between entities; and, detecting certain types of events.

We are in the process of moving KELVIN to encode the analysis output of all document-level tools for a document using Apache Thrift (Agarwal, Slee, and Kwiatkowski 2007). In addition to capturing the output of SERIF and FACETS we can include output from other NLP analysis tools, such as Stanford coreNLP and Apache OpenNLP.

For each entity with at least one named mention, we collect its mentions, the relations and events in which it participates, and all associated facets. Entities comprised solely of nominal or pronominal mentions are ignored for the Cold Start task, per the task guidelines.

Intra-Document Coreference. One option in our pipeline is to detect within-document entity chains that look problematic. For example, we have observed cases where family members or political rivals are mistakenly combined into a single entity cluster. This creates problems in knowledge base population where correct facts from distinct individuals can end up being combined into the same entity. For example, if Bill and Hillary Clinton are mentioned in a document that also mentions that she was born in the state of Illinois, a conjoined cluster might result in a knowledge base incorrectly asserting that Bill Clinton was born in Illinois. As an interim solution, we built a classifier to detect such instances and remove problematic clusters from further consideration in our pipeline, expecting that this might be a precision-enhancing operation.

Our classifier uses name variants from the American English Nickname Collection (LDC2012T11) and lightweight personal name parsing to identify acceptable variants (*e.g.,* Francis Albert Sinatra and Frank Sinatra). If our rules for name equivalence are not satisfied, then string edit distance is computed using a dynamic time warping approach to identify the least cost match; two entity mentions that fail to meet a closeness threshold by this measure are deemed to be mistakenly conflated. Organizations and geo-political entities are handled similarly. Name variants for geo-political entities (GPEs) include capital cites and nationalities for known countries. In addition, both are permitted to match with acronyms.

The document-level processing over the 26,000 newswire corpus found nearly 600,000 named entities (219,656 people (PERs), 174,189 GPEs and 200,612 organizations (ORGs)) with 3.6 million mentions and more than three million raw facts.

Cross-document Coreference Resolution

To build a coherent KB when information about an entity might be distributed across many documents, the results of the intra-document processing must be merged. At the very least, the entities must be clustered. To produce a high quality KB, duplicate relations must also be identified and merged (this is discussed further in the next section.

KELVIN uses an unsupervised, procedural clusterer called *Kripke* for entity clustering. *Kripke* is based on three principles: (1) coreferential clusters should match well in their names; (2) coreferential clusters should share contextual features; and (3) only a few discriminating contextual features should be required to disambiguate entities.

Kripke performs agglomerative clustering of document-level entities to produce cross-document entity clusters. To avoid the customary quadratic-time complexity required for brute-force pairwise comparisons, *Kripke* maintains an inverted index of names used for each entity. Only entities matching by full name or by some shared words or character n-grams are considered as potentially coreferential. Related indexing techniques are variously known as blocking (Whang et al. 2009) or canopies (McCallum, Nigam, and Ungar 2000).

total	% usable	relation
464472	5.1	org:stateorprovince_of_headquarters
358334	1.9	org:country_of_headquarters
244528	5.2	per:statesorprovinces_of_residence
188263	2.1	per:countries_of_residence
135991	100.0	gpe:part_of
16172	5.2	gpe:residents_of_stateorprovince
13926	100.0	per:top_member_employee_of
13926	100.0	org:top_members_employees
8794	7.6	per:stateorprovince_of_death
8038	5.2	per:stateorprovince_of_birth
6685	3.3	per:country_of_death
6107	2.1	per:country_of_birth
1561	100.0	per:employee_of
636	27.7	per:siblings
476	37.8	per:cities_of_residence
476	37.8	gpe:residents_of_city
356	58.4	per:other_family

Figure 2: Number of inferred relations and the percent meeting provenance requirements from a collection of 26K newswire articles.

Contextual matching only uses co-occurring named entities for support. Between two candidate clusters, the name variants of co-occurring entities are intersected. Each name is weighted by normalized inverse document frequency, so that rare, or discriminating names have a weight closer to 1. If the sum of the top ten weights exceeds a dynamic cutoff, then the contextual similarity is deemed to be adequate. With this technique the system can distinguish George Bush (41st U.S. president) from his son (43rd U.S. president), through co-occurring names (*e.g.,* Al Gore, Barbara Bush, and Kennebunkport for George H. W. Bush versus Dick Cheney, Laura Bush, and Crawford for George W. Bush).

A cascade of clustering passes is executed, during which name and contextual matching requirements are gradually relaxed. This allows higher precision matches to be made earlier in the cascade; these early merges assist more difficult merging decisions later on.

Inference over the Knowledge Base

By combining these document-level and cross-document technologies, a bare bones Cold Start system can be created. However, the application of several classes of inference can greatly improve the quality of the resulting KB. Because it actually produces a KB, the Cold Start task is more amenable to the use of inference than are the underlying technologies. KELVIN uses a suite of inferences, described in the following subsections.

Generating missing logical inverses. The presence of an assertion of [:homer, *per:children*, :bart] requires that the KB also contain an inverse relation of [:bart, *per:parents*, :homer]. While straightforward, this is an important inference. Relations are often extracted in only one direction

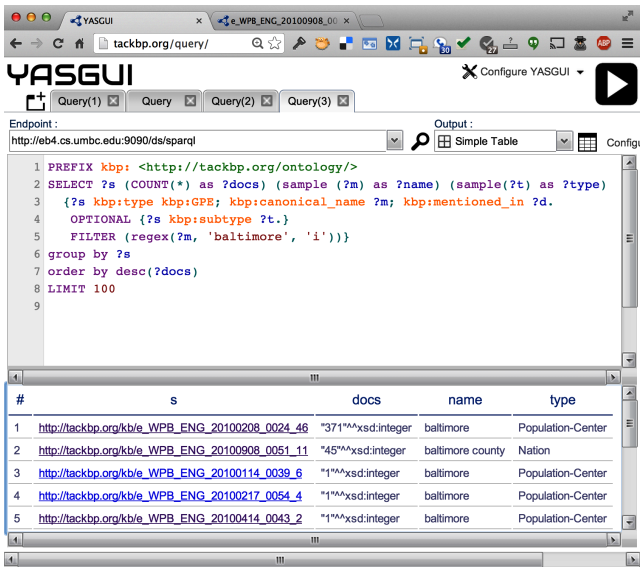

Figure 3: The RDF version of the extracted knowledge can be queried via SPARQL (here using the Yasgui interface) to find anomalies or collect data for analysis or training.

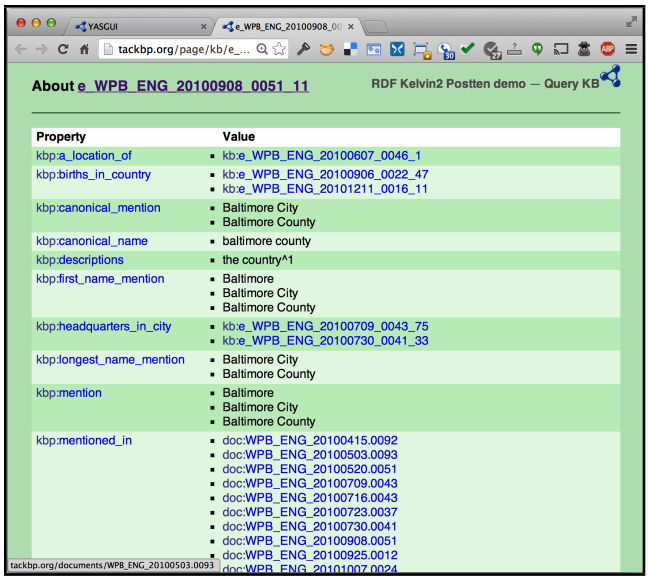

Figure 4: Pubby provides a simple way to browse the RDF version of the extracted knowledge via a Web browser.

during document-level analysis, yet both assertions should be explicit to aid with downstream inference.

Culling assertions. Inference can identify and discard implausible relations. For example, objects of predicates expecting a country (*e.g.,per:countries_of_residence*) must match a small, enumerable list of country names; Texas is not valid. Similarly, 250 is an unlikely value for a person's age. We validate certain slots to enforce that values must come from an accepted list of responses (*e.g.,* countries, religions), or cannot include responses from a list of known incorrect responses (*e.g.,* a girlfriend is not allowed as a slot fill for *per:other_family*).

Slot value consolidation. Extracting values for slots is a noisy process and errors are more likely for some slots than for others. The likelihood of finding incorrect values also depends the "popularity" of both the entity and slot in the collection. To reduce the number of false assertions, particularly for frequently appearing entities, slot values must be consolidated. This involves selecting the best value in the case of a single valued slot (*e.g.,per:city_of_birth*) and the best set of values for slots that permit more than one value (*e.g.,per:parents*). In both cases, KELVIN uses the number of attesting documents to rank candidate values, with greater weight given to values that are explicitly attested rather than inferred. For list-valued slots, it is difficult to know how many and which values to include. KELVIN makes the pragmatic choice to limit list-valued responses in a predicate-sensitive fashion, preferring frequently attested values. We associate two thresholds for selected list-valued predicates on the number of values that are reasonable – the first represents a number that is suspiciously large and the second is an absolute limit on the number of values reported. For example, KELVIN allows an entity to have at most eight values for *per:spouse*. To report more than three values, the additional spouses must be evidenced in multiple documents.

In our example collection, Barack Obama had 128 distinct values for employer, clearly an unreasonably large number. The number of attesting documents for each employer entity following a typical power law with large numbers of attesting documents for a few (e.g., 16 for *United States* and seven for *White House*) and most (e.g., *RNC*, *Red Cross*) found in just one document.

General Inference. The creation of a KB allows the use of a whole host of inferences on the triples of the KB. KELVIN applies about 25 forward chaining inference rules to increase the number of assertions in the KB. To facilitate inference of assertions in the Cold Start schema, it uses some unofficial slots that are subsequently removed prior to submission. For example, KELVIN adds slots for a person's sex, whether a person is deceased, and geographical subsumption (*e.g.,* Annapolis is part-of Maryland). The most prolific inferred relations involve family relationships, corporate management, and geo-political containment.

Many of these rules are logically sound and follow directly from the meaning of the relations. For example, two people are siblings if they have a parent in common and have an *other_family* relation if one is a grandparent of the other. Geographic subsumption produces a large number of additional relations, *e.g.,* knowing that a person's *city_of_birth* is Baltimore and that it is part of Maryland and that Maryland is a state supports the inference that the person's *stateorprovince_of_birth* is Maryland.

KELVIN also includes plausible rules that can infer, for example, that a person was a resident of a state or country if he attended a school located there. Some plausible rules have the form of *default rules*, *i.e.,* rules where one or more of the conditions is the absence of a fact. For example, KELVIN asserts that the sex of a male person's spouse

is female, but only if we do not already have a value for it. We assign a certainty factor between 0.0 and 1.0 to plausible rules and combine this with the certainty of facts to produce the conclusion's certainty.

The use of a small collection of inference rules can prove productive. Running KELVIN on the newswire collection produced about two million facts from which our inference rules generated an additional 1.5 million facts beyond the simple inverse facts. The geospatial inclusion inference rules were the most productive, accounting for nearly 90% of the new facts. However, nearly 85% of the inferred facts were unusable under the 2013 TAC-KBP provenance guidelines, which stipulate that relations must be attested in a single document. As an example, consider learning that Lisa is Homer's child in one document and that Bart is Homer's child in another. Assuming that the two Homer mentions corefer, it follows that Lisa and Bart are siblings. Nonetheless, to comply with the 2013 TAC task specification, KELVIN rejects any relation inferred from two facts unless one of the facts and both entities involved are mentioned in the same document. Figure 2 shows the number of relations inferred from the 26K news articles and the percentage that were usable given the 2013 Cold Start provenance requirements.

Parallelization of Inference. We conducted inference by loading the entire KB into memory, since in general, a rule might have any number of antecedent relations. However, many of our inference rules do not require arbitrary joins and can be run in parallel on KB subsets if we ensure that all facts about any entity are in the same subset. The fraction of rules for which this is true can be increased by refactoring them. For example, the rule for *per:sibling* might normally be written as

$$\text{X } \textit{per:parent} \text{ P} \wedge \text{Y } \textit{per:parent} \text{ P} \rightarrow \text{X } \textit{per:siblings} \text{ Y}$$

but it can also be expressed as

$$\text{P } \textit{per:child} \text{ X} \wedge \text{P } \textit{per:child} \text{ Y} \rightarrow \text{X } \textit{per:siblings} \text{ Y}$$

assuming that we materialize inverse relations in the KB (*e.g.,* asserting a child relation for every parent relation and vice versa). A preliminary analysis of our inference rules shows that all could be run in at most three parallelizable inference steps using a Map/Reduce pattern.

Development Tools

Evaluation is an essential step in the process of developing and debugging a KB population system that requires appropriate knowledge-base oriented techniques. We briefly describe several of the evaluation tools used by the KELVIN system as examples. Two were aimed at comparing the system's output from two different versions: *entity-match,* which focuses on differences in entities found and linked; and *kbdiff*, which identifies differences in relations among those entities. Together, these tools support assessment of relative KB accuracy by sampling the parts of two KBs that disagree (Lawrie et al. 2013). *Tac2Rdf* produces an RDF representation of a TAC KB supported by an OWL ontology and loads it into a standard triple store, making it available for browsing, inference and querying using standard RDF tools. *KB_Annotator* allows developers to browse the system

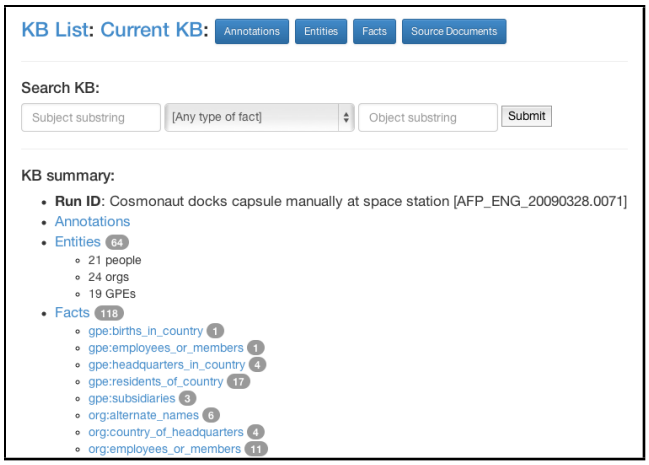

Figure 5: Overview of a KB including counts on entity type and relations.

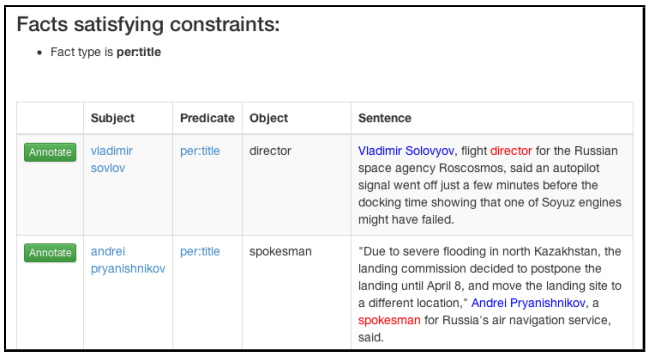

Figure 6: Results of a relation search.

output and annotate entities and relations as either supported or not by the document text provided as provenance.

Entity-match defines a KB entity as the set of its mentions. 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 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 KB. In the third case, the entity has gone undetected. The tool reports the entities and cases into which they fall. If there is disagreement between the KBs, it reports each corresponding entity in the second KB and the number of mentions that map to it.

Kbdiff identifies assertions in one KB that do not appear in the other. The challenge here is to identify which entities are held in common between the two KBs. Provenance is again useful; relations from different KBs are aligned if they have the same predicates and the provenance of their subjects and objects match. The algorithm works by first reading all the assertions in both KBs and matching them based on provenance and type. The output includes assertions in the first KB lacking a match in the second (prefixed by $<$) and those

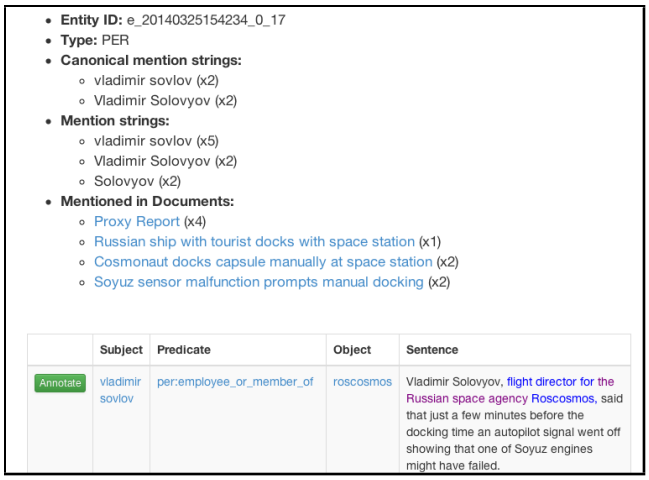

Figure 7: Entity view

in the second but not the first (prefixed by >.)

Tac2Rdf translates a KB in TAC format to RDF using an OWL ontology[1] that encodes knowledge about the concepts and relations, both explicit and implicit. For example, the Cold Start domain has an explicit type for geo-political entities (GPEs), but implicitly introduces disjoint GPE subtypes for cities, states or provinces, and countries through predicates like *city_of_birth*. Applying an OWL reasoner to this form of the KB detects various logical problems, e.g., an entity is being used as both a city and a country. The RDF KB results are also loaded into a triple store, permitting access by an integrated set of standard RDF tools including Fuseki for SPARQL (Prud'Hommeaux and Seaborne 2008) querying, Pubby for browsing, and the Yasgui SPARQL GUI.

Figure 3, for example, shows the results of an ad hoc SPARQL query for GPEs with the string "baltimore" in their canonical mention along with the number of documents in which they were mentioned and their subtype. Such queries are useful in identifying possible cross-document coreference mistakes (e.g., GPEs with mentions matching both *"X County" X*) and likely extraction errors (e.g., pairs of people connected by more than one relation in the set {*spouse*, *parent* and *other-family*}). Clicking on the second entity in the table of results opens the entity in the Pubby linked data browser, as shown in Figure 4.

The *KB_Annotator* system loads triples from a TAC submission into a KB and provides a Web-based interface allowing one to view a document's text, see the entities, entity mentions and relations found in it, and give feedback on their correctness. Figure 5 details an overview of the KB, by listing entities by their type with numeric counts overall and for each type. It also allows searching with subjects and objects found using string matches. Either all relations can be included or a particular relation. Search results are presented in a table as in shown in Figure 6. The light blue text represents links to entities and relationships. In this example, the object of the relation is a string rather than an entity, so the

[1] Available at http://tackbp.org/static/tackbp.ttl.

Figure 8: A document view where hovering over an entity mention reveals strings that are co-referent.

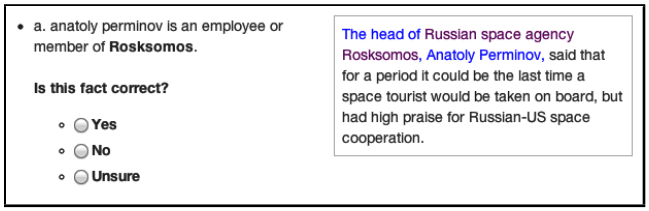

Figure 9: For each fact extracted from a document, a developer or assessor can express a judgment on its correctness.

object is not a link. While a relation is viewed as a list of facts, an entity consists of a type, canonical mentions, mention strings, documents in which the entity is mentioned, and facts pertaining to the entity. Figure 7 shows an example entity and Figure 8 shows document text with entity mentions identified in a contrasting color as well as the extracted facts. Mousing over a mention highlights it and any co-referential mentions.

Whenever a fact is encountered, the developer or assessor has the option of providing an annotation or judgment. As shown in Figure 9, the judgment is solicited by stating the relationship as a fact, showing the sentence with relevance parts of the text in contrasting colors, and a simple "yes," "no," "unsure" feedback on the fact. This allows an assessor to annotate errors in named entity resolution, in-document coreference, and relation detection with respect to particular facts. An annotator may alway annotate all the facts concerning a particular entity. Finally, Figure 10 shows a view of the annotations for a particular KB. These fact annotations can be used to error analysis and as training data.

Performance and challenges

Performance

KELVIN was initially developed for the 2012 Cold Start KBP evaluation and was the highest ranked system that year. A modified version of the was entered in 2013 and its best-performing run was ranked first. Reviewing the results shown in Table 1 for the five runs we submitted shows that there is considerable room for improvement.

KELVIN learned some interesting facts from a years worth of Washington Post articles (Gigaword 5th ed., LDC2011T07):

Figure 10: Developers and assessors can view the annotations concerning the facts in the KB.

- Harry Reid is an employee of the "Republican Party" and the "Democratic Party."
- Big Foot is an employee of Starbucks
- Steven Spielberg lives in Iran
- Jill Biden is married to Jill Biden

KELVIN also learned some true facts:

- Jared Fogle is an employee of Subway
- Freeman Hrabowski works for UMBC, founded the Meyerhoff Scholars Program, and graduated from Hampton University and the University of Illinois
- Supreme Court Justice Elena Kagan attended Oxford, Harvard and Princeton
- The Applied Physics Laboratory is a subsidiary of Johns Hopkins University
- Southwest Airlines is headquartered in Texas

Challenges

Although automated knowledge base construction from text is an area of active research (Carlson et al. 2010; Ji and Grishman 2011; Fan et al. 2012; Strassel et al. 2010; Freedman et al. 2011) it is still its infancy and faces many challenges. It requires multiple layers of language technology software, so advances in parsing, role labeling, entity recognition, anaphora resolution, relation extraction (Poon et al. 2010) and cross-document coreference resolution are fundamental to improving performance. Better techniques for information integration , probabilistic reasoning and exploiting background knowledge bases (Dong et al. 2014) are also important. We will briefly mention some of the issues that lie more at the KB end of the processing.

Large-scale KBs. Creating, updating and maintaining a large scale KB introduces many new challenges. Google's Freebase (Bollacker et al. 2008), for example, has over two billion triples with information on 50 million entities. Handling data at this scale requires the use of specialized databases that can efficiently store, index, query and retrieve data from persistent storage. Keeping such large KBs up-to-date means that we need to take an incremental, streaming approach to adding new information. This scenario introduces new challenges in reasoning to distinguish a change in the world from a change in our knowledge of the world. The current TAC Slot Filling task does not admit temporally qualified assertions and is thus temporally agnostic, *e.g.,* any current or former spouse is a valid fill.

Richer ontologies. The ontologies used in TAC and related efforts have remained quite simple, with O(10) types and O(100) relations. We need to be able to handle ontologies that are one or two orders of magnitude larger. In addition to being larger, the ontologies should also support richer representations that capture more knowledge and support more sophisticated reasoning. We must move beyond simple entities (PER, ORG, GPE) so as to model events, actions, states and situations.

Inference. We need better models and practical software systems for representing, managing and reasoning with probabilistic knowledge. Current approaches do not scale well, and we lack experience in integrating probabilistic reasoning across the analytic systems needed by a KBP system. Worse yet, the probability of introducing an inconsistency into the logic-oriented parts of a KB increases with its size. Any reasonably large KB supported by an expressive knowledge representation system is bound to contain contradictions, making some reasoning strategies untenable.

Provenance. The current model of provenance only allows one to justify a fact by pointing to strings in a document that assert it. Moreover, it simultaneously requires us to keep too much information (every fact must include provenance data) and too little (we keep just one justification for a fact). Future systems must be able to justify their beliefs using proof trees that are composed of attested facts, default assumptions, inferred facts and rules (logical and probabilistic). In some cases we may be able to use such justification trees to better manage a dynamic KB, *e.g.,* by knowing what else must change in our KB if a fact is retracted. We will also have to face the problem of when and how to gracefully "age out" and forget provenance data, lest it overwhelm our systems with information of low utility. At the same time provenance must be kept simple enough that an assessor can understand it to determine whether the system is making valid assertions.

Recovering from mistakes. Since a KB is at the core of the task, and also due to the fact that not every attestation of a relation must be identified, it is possible to aggressively allow low probability or inconsistent facts to be added to the KB at an early phase, for subsequent resolution. One particular way in which inconsistencies can arise is when multiple entities are mistakenly conjoined together; correctly declustering such entities can improve the accuracy in the KB.

Conclusion

One of the distinguishing characteristics of humans is that they are language users. Because of this, text documents

Run	P	R	F_1	P	R	F_1	P	R	F_1
1	0.5256	0.2843	0.3690	0.1620	0.2182	0.1859	0.2844	0.2551	0.2690
2	0.4929	0.3031	0.3753	0.1818	0.2406	0.2071	0.2969	0.2755	0.2858
3	0.4865	0.5000	0.4932	0.1849	0.3510	0.2423	0.3075	0.4342	0.3600
4	0.4799	0.5155	**0.4971**	0.1842	0.4168	**0.2555**	0.2950	0.4718	**0.3631**
5	0.4937	0.3053	0.3773	0.1453	0.2531	0.1846	0.2531	0.2823	0.2669

Table 1: Precision, recall, and F_1 for 0-hop slots (left columns), 1-hop slots (middle columns), and 0 and 1 hops (right columns).

of various kinds will continue to be one of the key sources of data for many domains in the foreseeable future. As information extraction systems continue to improve, they will become more useful and important as a way to extract, integrate and reason over data in the form of knowledge bases from text. This will not only produce more data for analysis, but will contribute to the creation and maintenance of background knowledge bases. These are important in supporting system that provide natural language question answering interfaces to datasets.

The JHU Human Language Technology Center of Excellence has participated in the TAC Knowledge Base Population exercise since its inception in 2009 and in Cold Start task since 2012. Recently we revised the initial KELVIN system by improving list-slot value selection, by developing a new, scalable approach to cross-document entity coreference, and through application of inference rules to discover facts not directly stated in the source text.

Acknowledgments

Partial support for this work was provided by NSF grants 0910838 and 1228673.

References

Agarwal, A.; Slee, M.; and Kwiatkowski, M. 2007. Thrift: Scalable cross-language services implementation. Technical report, Facebook.

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

Boschee, E.; Weischedel, R.; and Zamanian, A. 2005. Automatic information extraction. In *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 *24th Conf. on Artificial Intelligence*. AAAI.

Dong, X. L.; Murphy, K.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Strohmann, T.; Sun, S.; and Zhang, W. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM.

Fan, J.; Kalyanpur, A.; Gondek, D.; and Ferrucci, D. A. 2012. Automatic knowledge extraction from documents. *IBM Journal of Research and Development* 56(3.4):5–1.

Freedman, M.; Ramshaw, L.; Boschee, E.; Gabbard, R.; Kratkiewicz, G.; Ward, N.; and Weischedel, R. 2011. Extreme extraction: machine reading in a week. In *Conf. on Empirical Methods in Natural Language Processing*, 1437–1446. ACL.

Ji, H., and Grishman, R. 2011. Knowledge base population: Successful approaches and challenges. In *49th Meeting of the ACL: Human Language Technologies*, 1148–1158. ACL.

Lawrie, D.; Finin, T.; Mayfield, J.; and McNamee, P. 2013. Comparing and Evaluating Semantic Data Automatically Extracted from Text. In *Symposium on Semantics for Big Data*. AAAI Press.

Mayfield, J., and Finin, T. 2012. Evaluating the quality of a knowledge base populated from text. In *Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*. ACL.

McCallum, A.; Nigam, K.; and Ungar, L. 2000. Efficient clustering of high-dimensional data sets with application to reference matching. In *Knowledge Discovery and Data Mining*.

McNamee, P.; Stoyanov, V.; Mayfield, J.; Finin, T.; Oates, T.; Xu, T.; Oard, D. W.; and Lawrie, D. 2012. HLTCOE participation at tac 2012: Entity linking and cold start knowledge base construction. In *Text Analysis Conference (TAC)*.

McNamee, P.; Mayfield, J.; Finin, T.; Oates, T.; County, B.; Lawrie, D.; Xu, T.; and Oard, D. W. 2013. KELVIN: a tool for automated knowledge base construction. In *NAACL HLT*, volume 10, 32. (demo paper).

National Institute of Standards and Technology. 2008. Automatic content extraction 2008 evaluation plan (ACE08). Technical report.

Poon, H.; Christensen, J.; Domingos, P.; Etzioni, O.; Hoffmann, R.; Kiddon, C.; Lin, T.; Ling, X.; Mausam; Ritter, A.; Schoenmackers, S.; Soderland, S.; Weld, D.; Wu, F.; and Zhang, C. 2010. Machine reading at the University of Washington. In *1st Int. Workshop on Formalisms and Methodology for Learning by Reading*, 87–95. ACL.

Prud'Hommeaux, E., and Seaborne, A. 2008. SPARQL query language for RDF. Technical report, World Wide Web Consortium.

Strassel, S.; Adams, D.; Goldberg, H.; Herr, J.; Keesing, R.; Oblinger, D.; Simpson, H.; Schrag, R.; and Wright, J. 2010. The DARPA machine reading program. In *LREC*.

Text Analysis Conference. 2013. Proposed task description for knowledge-base population at tac 2013, english slot filling regular and temporal. http://bit.ly/1vOQ3CE.

Whang, S. E.; Menestrina, D.; Koutrika, G.; Theobald, M.; and Garcia-Molina, H. 2009. Entity resolution with iterative blocking. In *SIGMOD 2009*, 219–232. ACM.