HLTCOE Participation at TAC 2013

Paul McNamee
Johns Hopkins University
HLTCOE

Tim Finin
University of Maryland
Baltimore County

Dawn Lawrie
Loyola University Maryland

James Mayfield
Johns Hopkins University
HLTCOE

Abstract

The JHU HLTCOE participated in the Entity Linking and Cold Start Knowledge Base tasks in this year's Text Analysis Conference Knowledge Base Population evaluation. We have previously participated in TAC-KBP evaluations in 2009, 2010, 2011, and 2012. Our primary focus this year was on the Cold Start task; improvements to our existing KELVIN system included consolidating slot values for an entity, removal of suspect intra-document conference chains, streamlined cross-document entity coreference, and application of inference rules to expand the number of asserted facts.

1 Introduction

The JHU Human Language Technology Center of Excellence has participated in the TAC Knowledge Base Population exercise since its inception in 2009. Our focus over the past year was on the Cold Start task. We attempted to improve our KELVIN system (McNamee et al., 2012; McNamee et al., 2013) by improving list-slot value selection, cross-document entity coreference, and application of inference rules. We also made a last-minute submission to the English Entity Linking evaluation with a prototype cross-document entity coreference system, called *Kripke*.

2 Cold Start KB Construction

The TAC KBP 2012 Cold Start task is a complex task that requires application of multiple layers of NLP software. The most significant tool that we use is a NIST ACE entity/relation/event detection system, the BBN SERIF system. In addition to SERIF, significant components which we relied on include: a maximum entropy trained model for extracting personal attributes (FACETS, also a BBN tool); cross-document entity coreference (the COE KRIPKE system); and a procedurally implemented rule system.

2.1 System Description

KELVIN runs from two Unix shell scripts[1] 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 2013 Cold Start guidelines.

Our processing consists of the following steps, which are described in detail below:

1. Document-level processing
2. Curating intra-document coreference
3. Cross-document entity coreference
4. Generating missing logical inverses
5. Culling assertions that appear incorrect
6. Consolidating slot values for an entity
7. Applying inference rules to posit additional assertions

[1] Named Margaret and Fanny after Lord Kelvin's wives.

8. Again, generating missing assertions by producing logical inverses of existing facts

9. Post-processing steps

The *Margaret* script performs the document-level processing in parallel on our Sun Grid Engine computing cluster. *Fanny* executes the balance of the pipeline, and each of these steps is principally calculated as a single process.

2.1.1 Document-Level Processing

BBN's SERIF tool[2] (Boschee et al., 2005) 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,[3] 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 run each document through SERIF, and extract its annotations. Additionally we run another module named FACETS, described below, which adds some annotations about person entities. 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.

FACETS is an add-on package that takes SERIF's analyses and produces role and argument annotations about person noun phrases. 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 some role-specific attributes, such as employer for someone who has a job, (medical) specialty for physicians, or (academic) affiliation for someone associated with an educational institution.

2.1.2 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.[4] 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[5] 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 GPEs are handled similarly. Name variants for GPEs include capital cites and nationalities for known countries. In addition, both are permitted to match with acronyms.

2.1.3 Cross-document entity coreference

Last year we used the HLTCOE CALE entity linking system to assist with forming cross-document entity clusters, as is needed for the Cold Start task. This year we experimented with a new, more streamlined coreference tool called *Kripke*. We produced runs that used: (a) a normalized string matching baseline;

[2] Statistical Entity & Relation Information Finding

[3] `http://www.itl.nist.gov/iad/mig/tests/ace/2008/doc/ace08-evalplan.v1.2d.pdf`

[4] He was born in Arkansas.

[5] LDC2012T11

(b) *Kripke* with standard settings; and (c) *Kripke* using more aggressive clustering.

Kripke is an unsupervised, procedural clusterer that utilizes two principles: (a) to combine two clusters each must have good matching of both names and contextual features; (b) a small set of discriminating contextual features is sufficient for disambiguation. Additional details can be found in Section 3.1.

2.1.4 Generating missing logical inverses

Producing inverses is an entirely deterministic process that simply generates Y *inverse* X in *Doc D* from an assertion of X *slot* Y in *Doc D*. For example, inverse relations like per:parent and per:children, or per:schools_attended and org:students. While straightforward, this is an important step, as relations are often extracted in only one direction during document-level analysis, yet we want both assertions to be explicitly present in our KB to aid with downstream analysis.

2.1.5 Culling Assertions

Some assertions extracted from SERIF or FACETS can be quickly vetted for plausibility. For example, the object of a predicate expecting a country (*e.g.,* per:countries_of_residence) must match a small, enumerable list of country names; Massachusetts is not a reasonable response.[6] Similarly, 250 is an unlikely value for a person's age. We have procedures to check certain slots to enforce that values must come from a 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).

2.1.6 Consolidating Slot Values

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. For example, in processing a collection of 26K articles from the Washington Post, we observed more than fifty entities who had 14 or more employers. One entity was reported as having had 122 employers (per:employee_of)!

Slot value consolidation 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 can have more than one value (e.g, per:parents). In both cases, we use the number of attesting documents to rank candidate values, with greater weight given to values that were explicitly attested rather than implicitly attested via inference rules. See Figure 1 for the number of attesting documents for each of the values for the entity that have 122 distinct values for employer.

For slots that admit only a single value, we select the highest ranked candidate. However, for list-valued slots, it is difficult to know how many, and which values to allow for an entity. We made the pragmatic choice to limit list-values 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. Figure 1 shows the thresholds we used for some predicates. For predicates in our table, we accepted the nth value on the candidate list if n did not exceed the first threshold and rejected it if n exceeded the second. For n between the thresholds, a value is accepted only if it has more than one attesting document.

2.1.7 Inference

We apply a number of forward chaining inference rules to increase the number of assertions in our KB. To facilitate inference of assertions in the Cold Start schema, we introduce some unofficial slots into our KB, which are subsequently removed prior to submission. For example, we add slots for the sex of a person, and geographical subsumption (*e.g.,* Gaithersburg is part-of Maryland). The most prolific inferred relations were based on rules for family relationships, corporate management, and geopolitical containment.

Many of the 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 two people have an "other_family" relation if they one is a grandparent of the other. Our knowledge of geographic subsumption produced a

[6] In 2013, neither is Texas.

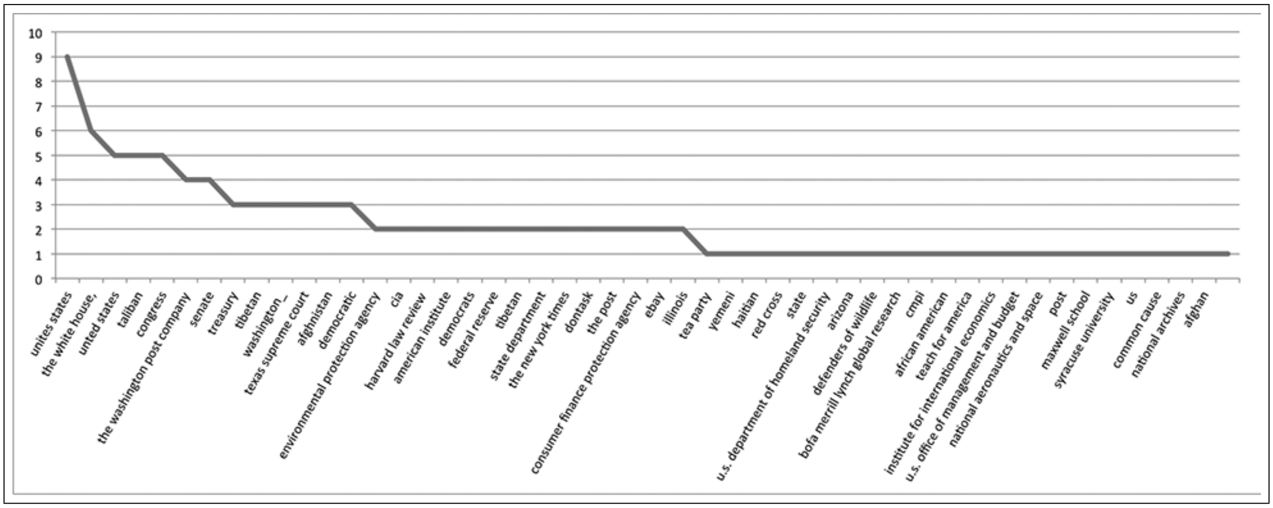


Figure 1: After processing 26,000 news articles from the Washington Post, the largest cluster for President Barack Obama had 128 distinct values for employer. The number of attesting documents for each followed a typical power law, with nine documents for the most popular value only one for the majority. This figure shows the distribution for the first 50.

large number of additional relations, e.g., knowing that a person's *city_of_birth* is Gaithersburg and that it is part of Maryland and that Maryland is a state supports the inference that the person's *state-orprovince_of_birth* is Maryland.

We aim for high precision, but do not require 100% soundness in all of our rules. For example, we infer that if a person attended a school S, and S has headquarters in location L, then the person has been a resident of L.

In general, we do not add an inferred fact that is already in our knowledge base. Some of the rules are default rules in that they only add a value for a slot for which we have no values. For example, we know that person P1 is the spouse of person P2 and that the sex of P1 is male and we have no value for the sex of P2, we infer that P2 is female. In this case, the rule is both a default rule and one whose conclusion is very often, but not always, true.

relation	**T1**	**T2**
per:children	8	10
per:countries_of_residence	5	7
per:employee_of	8	10
per:member_of	10	12
per:parents	5	5
per:religion	2	3
per:schools_attended	4	7
per:siblings	9	12
per:spouse	3	8

Table 1: The number of values for some multi-valued slots were limited by a heuristic process that involved the number of attesting documents for each value and two thresholds.

The current TAC-KBP guidelines stipulate that relations must be attested in a single document, which notably constrains the number of inferred assertions which we are permitted to make. Therefore, we filter any relations not evidenced entirely in a single document prior to submission. As an example of a valid inference we filtered out, 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 co-refer, it follows that Lisa and Bart are siblings. The heuristic filter we used rejected any relation inferred from two facts unless one of the facts and both entities involved were mentioned in the same document.

Figure 2 shows the the number of additional relations that were inferred from the facts extracted from a collection of 26K Washington Post articles for which 140751 entities were found. For each we also show what percent were usable given the Cold-Start provenance requirements.

We ran the inference step over the entire knowledge base which had been loaded into memory, since in general, a rule might have any number of an-

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
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

Table 2: The number of of inferred relations and the percent that met the provenance requirements from a collection of 26K Washington Post articles.

Name	Clustering	Inference	InDoc	Extra
hltcoe1	Exact			
hltcoe2	*Kripke*			
hltcoe3	*Kripke*	Yes		
hltcoe4	*Kripke*	Yes	Yes	
hltcoe5	*Kripke*	Yes		Yes

Table 3: Description of conditions for HLTCOE Cold Start runs.

	0-hop			1-hop		
Name	P	R	F	P	R	F
hltcoe1	**0.429**	0.267	0.329	0.072	0.109	0.087
hltcoe2	0.410	0.361	**0.384**	0.084	0.113	0.097
hltcoe3	0.350	0.278	0.310	0.082	0.124	0.098
hltcoe4	0.405	0.327	0.362	**0.214**	0.110	**0.145**
hltcoe5	0.354	**0.390**	0.371	0.076	**0.131**	0.096

Table 4: Precision, Recall, and F_1 scores for our submitted runs using the Grishman v2 scorer.

tecedent relations. However, we realized that many of our inference rules do not require arbitrary joins and could be run in parallel on subsets of the knowledge base 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

X per:parent P $\wedge$ Y per:parent P $\rightarrow$ X per:siblings Y

but can also be expressed as

P per:child X $\wedge$ P per:child Y $\rightarrow$ X per:siblings Y

assuming that we materialize inverse relations in the knowledge base (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.

2.1.8 Post-processing

The final steps in our pipeline ensure compliance with the task guidelines. We normalize temporal expressions, ensure that all entities have mentions, insist that relations are consistent with the types of their subjects and objects, and we confirm that logical inverses are asserted, and so forth.

2.2 Submitted Runs

We submitted five experimental conditions that started with a simplistic baseline pipeline, and which used (or didn't use) *Kripke* cross-document entity coreference, inference rules, within-document mention-chain purification, and corpus augmentation. Table 3 summarizes the various conditions.

The only of our runs which made any direct use of external resources was hltcoe5, which is just like hltcoe3 (+*Kripke*, +inference) except that it was run on the ColdStart '13 corpus with 50k New York Times articles also mixed in to see if the additional documents aid cross-document clustering and slot consolidation.[7] KELVIN does not access the Internet during processing.

The number of times each slot was asserted for run hltcoe5 is given in Table 5.

Table 6 lists the number of entities of each type which are included in each of our runs. Note that as entities having no asserted relations cannot improve scores in the ColdStart task, we did not include such "mention only" entities in our submissions. The number of reported entities is generally similar in each run, with differences likely attributable to changes in cross-document entity coreference.

In Table 7 the number of facts asserted for each experimental condition is broken down by entity type for each submitted run.

[7] Clearly some of our components use linguistics resources such as parsers or supervised NER modules based on annotated corpora.

Slot name	# Assertions
per:employee_or_member_of	44339
org:alternate_names	39432
org:employees_or_members	36993
per:statesorprovinces_of_residence	30309
gpe:residents_of_stateorprovince	30309
per:title	20377
per:countries_of_residence	10954
gpe:residents_of_country	10954
per:cities_of_residence	7943
gpe:residents_of_city	7943
gpe:employees_or_members	7346
per:top_member_employee_of	5644
org:top_members_employees	5644
org:parents	5051
org:city_of_headquarters	4863
gpe:headquarters_in_city	4863
org:stateorprovince_of_headquarters	4502
gpe:headquarters_in_stateorprovince	4502
per:alternate_names	4480
per:origin	4170
org:country_of_headquarters	3438
gpe:headquarters_in_country	3438
org:subsidiaries	3012
per:spouse	2685
per:country_of_birth	2511
gpe:births_in_country	2511
gpe:subsidiaries	2039
per:date_of_death	1828
per:age	1701
per:parents	1390
per:children	1390
per:schools_attended	1285
org:students	1285
per:siblings	1123
per:charges	870
org:founded_by	839
per:organizations_founded	741
per:other_family	676
org:members	363
per:date_of_birth	301
org:date_founded	261
org:member_of	253
per:stateorprovince_of_death	227
gpe:deaths_in_stateorprovince	227

Table 5: Number of assertions for each predicate for run hltcoe5. Slots not listed, were never asserted.

Run	PER	ORG	GPE	Total
hltcoe1	121,934	66,154	13,141	201,229
hltcoe2	113,159	62,179	13,640	188,978
hltcoe3	112,496	61,887	13,548	187,931
hltcoe4	111,718	60,472	13,497	185,687
hltcoe5	119,940	65,765	14,316	200,021

Table 6: Number of entities identified in the evaluation corpus for each run.

Run	PER	ORG	GPE	Total
hltcoe1	99,213	90,070	31,734	221,017
hltcoe2	94,175	85,355	30,303	209,833
hltcoe3	112,128	92,964	46,193	251,285
hltcoe4	103,262	76,122	43,148	222,532
hltcoe5	145,673	106,159	74,828	326,660

Table 7: Numbers of facts, by entity type for each run.

2.3 Discussion

Comparing our various experimental conditions, we make the following observations.

It appears that use of *Kripke* cross-document coreference does improve recall, as was expected; 0-hop recall rises from 0.267 in hltcoe1 to 0.361 in hltcoe2. Precision is hardly affected, and thus F_1 rises.

Use of inference rules (contrast hltcoe3 to hltcoe2) appears to hurt performance of 0-hop queries. To date we have not currently analyzed the reasons for this, but we conjecture that the requirement to support all asserted facts from evidence in a single document may have been a cause. Curiously, 1-hop performance was not degraded.

Eradicating spurious within-document mention chains (hltcoe4 vs. hltcoe3) did notably improve precision and recall for both types of queries. The gain in precision was hoped for, however, the gain in recall was not something that we had predicted. We suspect the boost in recall is due to improved cross-document clustering decisions aided by fewer errors caused by within-document coreference decisions.

Finally, augmenting our corpus with a comparable size of news documents improved recall (both 0-hop and 1-hop). While facts learned solely from the expansion documents would have to be deleted, this may have helped us select among multiple choices for slot values observed in the evaluation documents, and may also have aided in cross-document coreference decisions among entities.

3 English Entity Linking

Our approach to the entity linking task was to use the *Kripke* tool for cross-document entity coreference resolution to form clusters. We did not use the TAC-KBP KB, except to extract a list of KBIDs and their corresponding Wikipedia titles. However, we did process a dump of DBpedia, for which we could map many entities to English Wikipedia, and thereby to the TAC-KBP KB identifiers.

3.1 Kripke: a tool for cross-document coreference

The *Kripke* system[8] takes a set of document-level entities and performs agglomerative clustering on them to produce cross-document entity clusters. The tool is written in approximately 2000 lines of Java source code. The intent is for the system to have a precision bias, which we feel is appropriate for knowledge base population.

The principles on which *Kripke* operations are:

- Coreferential clusters should match well in their names.
- Coreferential clusters should share contextual features.
- Only a few, discriminating contextual features should be required to disambiguate entities.

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 some shared words or character n-grams are considered as potentially coreferential.[9] Related indexing techniques are variously known as blocking (Whang et al., 2009) or canopies (McCallum et al., 2000).

At present, contextual matching is accomplished solely by comparing named entities that co-occur in the same document. Between candidate clusters, the sets of all names occurring in any document forming each cluster are intersected. Each name is weighted by normalized Inverse Document Frequency, so that rare, or discriminating names have a weight closer to 1. The top-k (*i.e.,* k=10) weighted names were used, and if the sum of those weights exceeds a cutoff, then the contextual similarity is deemed adequate. Such a technique should be able to tease apart George Bush (41st president) and his son (43rd president) through co-occurring names (*e.g.,* Al Gore, Barbara Bush, Kennebunkport, James Baker versus the entities Dick Cheney, Laura Bush, Crawford, Condolezza Rice).

The system runs by executing a cascade of clustering passes, where in each subsequent pass conditions are relaxed in the requirements for good name and contextual matching. The hope is that higher precision matches are made in earlier phases of the cascade, and these will facilitate more difficult matches later on.

3.2 English runs

Kripke was principally designed to do clustering, and thus is more suited to the NIL clustering aspect of the entity linking evaluation than linking to the TAC-KBP knowledge base. To make some attempt to link to the KB, we created a surrogate document representation for TAC-KBP KB entities found in DBpedia. For each of these documents, the seed or “focal” entity from which it was generated, is known to be linked to the corresponding TAC-KBP KB entity. The names in the surrogate documents come from names found in relationships within DBPedia (e.g., family members, birthplaces, employers, etc...).

We submitted two entity linking runs[10]. The first, hltcoe1, only performed NIL clustering, and did not use the TAC-KBP KB in any way. The second, hltcoe2, used the surrogate documents to attempt links to KB entities (*i.e.,* non-NILs) as well.

We used the following process on data from DBpedia (Bizer et al., 2009) to create triples representing the surrogate documents. For each entity (PER, ORG or GPE) in the TAC-KBP KB we found all of the other entities to which it was related in the DBpedia dataset, resulting in about 4.6 million unique entity pairs. For example, the entity *Alan_Turing* has eleven related entities: *Alonzo_Church*, *Cheshire*, *Government_Communications_Headquarters*, *King's_College,_Cambridge*, *Maida_Vale*, *Princeton_University*, *Robin_Gandy*, *Royal_Society*, *University_of_Cambridge*, *University_of_Manchester* and *Wilmslow*. For each of the related entities, we created mention strings from the DBpedia data using any of nine properties associated with names (e.g., rdf:label, foaf:name, foaf:givenName, dbpo:birthName and dbpo:alias).

[8] Named after Princeton philosopher Saul Kripke, who wrote a book on naming entities in the 1970s.

[9] Support for orthographically dissimilar name variants (*i.e.,* aliases) was planned, but not implemented in time for this year.

[10] By TAC convention, these run names appear similar to our Cold Start runs described earlier, but are wholly unrelated runs.

Name	All	in KB	Absent	News	Web	Diss.	PER	ORG	GPE
hltcoe1	0.329	0.000	0.656	0.389	0.429	0.160	0.382	0.500	0.116
hltcoe2	0.323	0.028	0.615	0.395	0.375	0.160	0.374	0.477	0.126

Table 8: B^3+ F_1 scores reported by NIST for various types of queries (all queries, those in the KB, not in the KB; from newswire documents, from web pages, from discussion forums; for person entities, organizations, and geo-political entities).

```
:9462264_TAC2009KB_E0769190 canonical_mention ”Alan Turing” 9462264_DOC 7 8 1.0
:9462264_TAC2009KB_E0769190 type per
:9462264_TAC2009KB_E0769190 mention ”Alan Mathison Turing” 9462264_DOC 11 12 1.0
:9462264_TAC2009KB_E0769190 mention ”Alan Turing” 9462264_DOC 7 8 1.0
:9462264_TAC2009KB_E0769190 mention ”Turing, Alan Mathison” 9462264_DOC 9 10 1.0
:9462265_TAC2009KB_E0767314_NONFOCAL mention ”Alonzo Church” 9462264_DOC 15 16 1.0
:9462265_TAC2009KB_E0767314_NONFOCAL mention ”Church, Alonzo” 9462264_DOC 13 14 1.0
:9462266_DO_NOT_CLUSTER mention ”King’s College” 9462264_DOC 21 22 1.0
:9462266_DO_NOT_CLUSTER mention ”King’s College, Cambridge” 9462264_DOC 17 18 1.0
:9462266_DO_NOT_CLUSTER mention ”King’s College of our Lady and Saint Nicholas” 9462264_DOC 19 20 1.0
:9462271_TAC2009KB_E0241809_NONFOCAL mention ”Chomsky, Avram Noam” 9462264_DOC 43 44 1.0
:9462271_TAC2009KB_E0241809_NONFOCAL mention ”Noam Chomsky” 9462264_DOC 41 42 1.0
:9462279_DO_NOT_CLUSTER mention ”Stevan Harnad” 9462264_DOC 71 72 1.0
:9462281_TAC2009KB_E0273935_NONFOCAL mention ”Ned Block” 9462264_DOC 79 80 1.0
```

Figure 2: These TAC assertions are part of the surrogate document generated from the DBPedia data for the Wikipedia entity *Alan_Turing* used in the entity linking process. The first entity *(:9462264_TAC2009KB_E0769190)* is the focus of the document, those tagged with NONFOCAL are related TAC-KBP KB entities and those tagged with *DO_NOT_CLUSTER* are related entities not in the TAC-KBP KB.

Entity identifiers were created for these related entities and tagged with a string indicating that they were also TAC-KBP KB entities or or entities not in the KB. A surrogate document for the entity was then generated as a set of TAC assertions including its mentions as well as the mentions for the related entities. Figure 2 shows a portion of the 38 triples produced for the surrogate document for *Alan_Turing*.

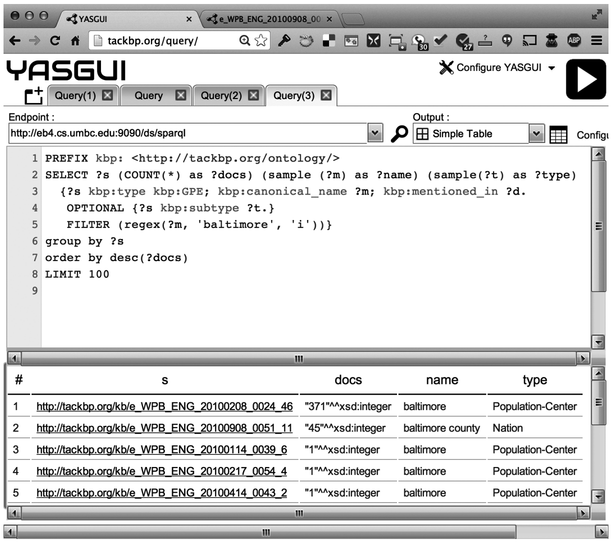
The B^3+ (modified B-cubed) F_1 scores for hltcoe1 and hltcoe2 from the TAC KBP 2013 Notebook are given in Table 8.

Performance is poor for within-KB entities, even in hltcoe2 where an attempt was made to map to the TAC-KBP KB. Compared to newswire, scores are notably degraded on Web page and Discussion Fora queries, and queries about persons and organizations outperform those for geo-political entities.

4 Development Tools

We created several software tools to support our development of our 2013 Cold Start system. 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. *Tac2Rdf* produces an RDF representation of a TAC knowledge base and loads it into a standard triple store making it available for browsing, inference and querying using standard RDF tools.

Entity-match 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



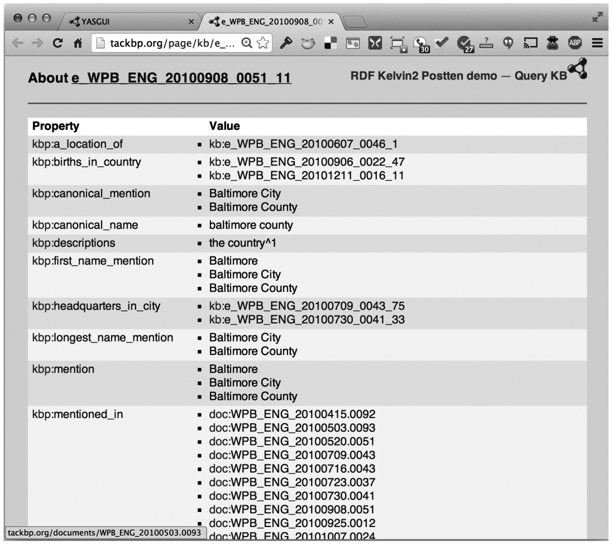
```
PREFIX kbp: <http://tackbp.org/ontology/>
SELECT ?s (COUNT(*) as ?docs) (sample (?m) as ?name) (sample(?t) as ?type)
  {?s kbp:type kbp:GPE; kbp:canonical_name ?m; kbp:mentioned_in ?d.
   OPTIONAL {?s kbp:subtype ?t.}
   FILTER (regex(?m, 'baltimore', 'i'))}
group by ?s
order by desc(?docs)
LIMIT 100
```

#	s	docs	name	type
1	http://tackbp.org/kb/e_WPB_ENG_20100208_0024_46	"371"^^xsd:integer	baltimore	Population-Center
2	http://tackbp.org/kb/e_WPB_ENG_20100908_0051_11	"45"^^xsd:integer	baltimore county	Nation
3	http://tackbp.org/kb/e_WPB_ENG_20100114_0039_6	"1"^^xsd:integer	baltimore	Population-Center
4	http://tackbp.org/kb/e_WPB_ENG_20100217_0054_4	"1"^^xsd:integer	baltimore	Population-Center
5	http://tackbp.org/kb/e_WPB_ENG_20100414_0043_2	"1"^^xsd:integer	baltimore	Population-Center

Figure 3: The RDF version of the extracted knowledge can be queried via SPARQL, here using the Yasgui interface.

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 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.

Kbdiff was inspired by the Unix *diff* utility and identifies 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 and matching them based on provenance and type. for each 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 and those that do not match one from the first KB are output preceded by a ">".



About e_WPB_ENG_20100908_0051_11

RDF Kelvin2 Postten demo — Query KB

Property	Value
kbp:a_location_of	kb:e_WPB_ENG_20100607_0046_1
kbp:births_in_country	kb:e_WPB_ENG_20100906_0022_47 kb:e_WPB_ENG_20101211_0016_11
kbp:canonical_mention	Baltimore City Baltimore County
kbp:canonical_name	baltimore county
kbp:descriptions	the country^1
kbp:first_name_mention	Baltimore Baltimore City Baltimore County
kbp:headquarters_in_city	kb:e_WPB_ENG_20100709_0043_75 kb:e_WPB_ENG_20100730_0041_33
kbp:longest_name_mention	Baltimore City Baltimore County
kbp:mention	Baltimore Baltimore City Baltimore County
kbp:mentioned_in	doc:WPB_ENG_20100415.0092 doc:WPB_ENG_20100503.0093 doc:WPB_ENG_20100520.0051 doc:WPB_ENG_20100709.0043 doc:WPB_ENG_20100716.0043 doc:WPB_ENG_20100723.0037 doc:WPB_ENG_20100730.0041 doc:WPB_ENG_20100908.0051 doc:WPB_ENG_20100925.0012 doc:WPB_ENG_20101007.0024

tackbp.org/documents/WPB_ENG_20100503.0093

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

Tac2Rdf translates a knowledge base in TAC format to RDF using a simple OWL ontology [11]. The results are then loaded into the Jena triple store using the TDB store, permitting access by an integrated set of standard RDF tools including the Fuseki SPARQL (Prud'Hommeaux and Seaborne, 2008) server for querying, Pubby (Prud'Hommeaux and Seaborne, 2008), for browsing the knowledge base and the Yasgui SPARQL GUI. Figure 3, for example, shows the results of an ad hoc SPARQL query that shows GPE entities with the string "baltimore" in their canonical mention along with the number of documents in which they were mentioned and their subtype, if one was extracted. Clicking on the second entity in the table of results opens the entity in the Pubby linked data browser, as shown in Figure 4.

5 Conclusion

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. We improved the KELVIN system developed for the 2012 Cold Start task by improving list-slot value selection, cross-

[11] Available at http://ebiquity.umbc.edu/ontologies/tackbp/-2012/tackbp.ttl"

document entity coreference, and application of inference rules. We also used it to make a submission to the English Entity Linking evaluation with a prototype cross-document entity coreference system, called *Kripke*.

6 Acknowledgments

We are grateful to BBN for providing SERIF and supporting our work with it.

References

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. Dbpedia-a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165.

E. Boschee, R. Weischedel, and A. Zamanian. 2005. Automatic information extraction. In *Proceedings of the 2005 International Conference on Intelligence Analysis, McLean, VA*, pages 2–4.

Dawn Lawrie, Tim Finin, James Mayfield, and Paul McNamee. 2013. Comparing and Evaluating Semantic Data Automatically Extracted from Text. In *AAAI 2013 Fall Symposium on Semantics for Big Data*. AAAI Press, November.

Andrew McCallum, Kamal Nigam, and Lyle Ungar. 2000. Efficient clustering of high-dimensional data sets with application to reference matching. In *Knowledge Discovery and Data Mining (KDD)*.

Paul McNamee, Veselin Stoyanov, James Mayfield, Tim Finin, Tim Oates, Tan Xu, Douglas W. Oard, and Dawn Lawrie. 2012. HLTCOE participation at TAC 2012: Entity linking and cold start knowledge base construction. In *Text Analysis Conference (TAC)*, Gaithersburg, Maryland, November.

Paul McNamee, James Mayfield, Tim Finin, Tim Oates, Baltimore County, Dawn Lawrie, Tan Xu, and Douglas W Oard. 2013. KELVIN: a tool for automated knowledge base construction. In *Human Language Technologies: The 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, volume 10, page 32.

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

Steven Euijong Whang, David Menestrina, Georgia Koutrika, Martin Theobald, and Hector Garcia-Molina. 2009. Entity resolution with iterative blocking. In *SIGMOD 2009*, pages 219–232. ACM.