Abstract

The JHU HLTCOE participated in the Cold Start and the Trilingual Entity Linking and Discovery tasks of the 2015 Text Analysis Conference Knowledge Base Population evaluation. For our fourth year of participation in Cold Start we continued our research with the KELVIN system. We submitted experimental variants that explore use of linking to Freebase and adding additional relations. This is our first year of participation in EDL. We used KELVIN in three runs and experimented with an alternate system for named entity recognition and linking for two additional runs.

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 developing our KELVIN system (McNamee et al., 2012; McNamee et al., 2013; Mayfield et al., 2014; Finin et al., 2014) as a core technology for multiple TAC tasks. This year we used it in our participation of the Cold Start and the Trilingual Entity Linking and Discovery (TEDL) tasks.

This is the fourth year that KELVIN participated in the Cold Start task. This year we enhanced our system by linking entities to Freebase to both improve our choice of canonical mentions and entity merging, augmented our relation extraction capabilities using pattern matching and an open information extraction system, and improved inference and entity merging through a variety of software engineering and architectural modifications. We adapted KELVIN for use in the TEDL task and also applied an independent system for several submitted runs.

In the rest of the paper we present our systems, which are architecturally similar to our 2014 submission, the additional components required for the Trilingual Entity Linking and Discovery task, and briefly discuss our experimental results.

2 Cold Start KB Construction

The TAC-KBP 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, BBN SERIF (Ramshaw et al., 2011). SERIF provides a substrate that includes entity recognition, relation extraction, and within-document coreference analysis. 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 HLTCOE Kripke system); and a procedurally implemented rule system.

KELVIN is organized as a pipeline with three
stages: (i) document level processing done in parallel on small batches of documents, (ii) cross-document co-reference resolution to produce an initial KB, and (iii) knowledge-base enhancement and refinement through inference and relation analysis. The next section describes the major steps in these stages.

3 Cold Start 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 Cold Start guidelines.

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

1. Document-level processing
2. Extended Document-level processing
3. Cross-document entity coreference
4. KB cleanup and slot value consolidation
5. Linking entities to an external background KB
6. Applying inference rules to posit additional assertions
7. KB-level entity clustering
8. KB cleanup and slot value consolidation
9. Selecting the best provenance metadata
10. Post-processing

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.\(^4\) Additionally we run another module named FACETS, described below, which adds attributes about person entities. For each entity with at least one named mention, we collect its mentions, the relations, and events in which it participates. Entities comprised solely of nominal or pronominal mentions are ignored for the Cold Start task, per the task guidelines. Finally, the output from each document is entered into a Concrete (Ferraro et al., 2014) object, which is our standard representation for information extracted from a document.

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.

3.2 Extended Document-Level Processing

This section describes the five additional steps taken once SERIF and FACETS are run. These generally address short comings in the tools or add additional information that was not found by the primary tools.

---

\(^1\)Named Margaret and Fanny after Lord Kelvin’s wives.
\(^2\)Statistical Entity & Relation Information Finding

\(^3\)http://www.itl.nist.gov/iad/mig/tests/ace/2008/doc/ace08-evalplan.v1.2d.pdf

\(^4\)We used an in-house version of SERIF, not the annotations available from LDC.
3.2.1 Relation Extraction by Pattern Matching

The patterns focused on relations that were either not attested or were observed to be frequently missed. They included four different patterns.

The first pattern looked for occurrences of `<PER>` of `<GPE>`. When this pattern was found, the relation per:countries_of_residence, per:statesorprovinces_of_residence, or per:cities_of_residence was attested, depending on the GPE subtype identified by SERIF. Patterns where the GPE was of the NATION_SUBTYPE were considered per:countries_of_residence relations. Patterns where the GPE was of the STATE_SUBTYPE were considered per:statesorprovinces_of_residence relations, and patterns where the GPE was of the CITY_SUBTYPE were considered per:cities_of_residence. If the subtype was something else, than the relation would not be asserted. This pattern resulted in roughly twenty-five more per:countries_of_residence assertions. This was a 0.3% increase in the number of assertions for this relation.

The second pattern focused on the relation org:politicalreligious_affiliation by looking for the pattern JJ `<ORG>` where that word in that parse tree labeled as JJ was either a religion, the word “Republican”, or the word “Democratic.” A religion was identified by a white list of religions. In addition words that were identified as semantically similar to Republican and Democratic were also considered acceptable. Semantically similar phrases were ones that scored higher than 0.7 by the STS service (Kashyap et al., 2016). This added roughly 5,000 assertions about the political or religious affiliations of organizations. No relations of this type had been identified by SERIF/FACETS.

The third pattern identified sentences containing an organization and a URL. URLs were identified by locating strings containing the characters “www.” and “http”. The core fragment of the url was identified. A fuzzy string match was used to find the highest scoring organization in the sentence. Finally, the fuzzy match score must be above a certain threshold in order to be added. Close to one thousand org:website were asserted using these rules. No relations of this type had been identified by SERIF/FACETS.

The final pattern discovered per:age relations by looking for `<PER>` followed somewhere in the sentence by the word “age” and a number. This process identified roughly 150 additional per:age relations, a 25% increase for this relation.

3.2.2 Relation Extraction using Open IE

Open information extraction is an alternative method of information extraction. An open information extraction system extracts a greater diversity of relations that may or may not align to TAC relations or to entities that have previously been identified. Given that multiple IE approaches are used, the output of the systems must be aligned. In addition to this challenge, the more free-form relations coming from Open IE must be aligned to TAC relations. In general, rather than using a rule based approach to determine the translation, a more automated approach was adopted that bootstrapped from relations identified by both systems and the integration of semantic similarity were used.

In more detail the Ollie system (Mausam et al., 2012) was run over each document in the collection. First, all extractions that did not include at least one known mention were eliminated. Then known mentions were mapped to their entities as identified by SERIF. Finally, the relation was established by comparing the text expressing the relation to a list of at most thirty different ways of expressing that relation. The best match above a threshold was used to assert a relation. In the case of relations between an entity and a string, rather than between entities, the argument was also examined to assure that the string had characteristics associated with that slot. For instance, if the slot was per:cause_of_death, the the fill had to come from a list of 625 words that would cause death.

In order to determine ways of expressing relations, a run on 26,000 Washington Post articles was used. The primary assumption made was that if a relation between two entities had already been observed by SERIF or FACETS in a particular sentence, then the relation found by SERIF/FACETS was the same one as the expressed by the text extracted by the Open IE system. A similar technique was used for slots with string values. There were a few instances where no prior examples of the slot

---

5 Due to a programming error neither per:statesorprovinces_of_residence or per:cities_of_residence were added.
existed. In this case a few ways of expressing the relationship were hand-crafted and used along side the ones that were automatically identified.

3.2.3 Refining Canonical Mentions
The default method of determining the canonical mention was to use the longest string identified. This caused two problems. One came from the fact that clauses were frequently included as parts of the name. Other errors came from errors in within document co-reference, where a minority entity was chosen for the canonical mention. An example of an error caused by this rule was choosing “whoever rules North Korea” as the canonical mention of an entity where every other mention is “North Korea.”

The refined canonical mention favors the most frequent name in the mention chain that is associated with a Freebase entity. In the event that there is no Freebase entity, the algorithm defaults to the most frequent mention. There are a few caveats to the algorithm. The first is that single names for people are avoided. In addition abbreviations are avoided when the expansion also appears. In the event that no candidates for canonical mentions exists, the algorithm defaults to the longest string. At the document level this leads to 64,384 changes in canonical mentions, which effects 1% of the document entities. This change has a ripple effect because Kripke, the cross-document conference system, is highly dependent on the canonical names of the entities. The new canonical names lead to an 3.5% reduction in the number of entities for the TAC 2015 document set.

3.2.4 Revisiting Dates
Although SERIF reports dates in TIMEX2 format, these dates are not always compliant with the TAC guidelines. For instance, some of the dates are written with only the year (i.e., “1948” rather than “1948-XX-XX”). In addition some of the dates are relative rather than absolute dates, for instance, “PXD” for “a few days ago.” These need to be revised based on the date the document was published. The revision of dates effected 2,295 dates. Our system initially found 2,841 dates. This means that 81% were changed. Another 243 or 9% were removed because there was insufficient information to determine the actual date in the format required to comply with the TAC guidelines.

3.2.5 Augmenting Entity Mentions
It was determined that SERIF overlooks entities that appear in the headline and the dateline. This is particularly problematic because many past entry points chosen by the evaluators were from the headline.

The process to add mentions of entities from the headline relied on matching the strings in the headlines to already established co-reference chains in the body of the document. The dateline may also contain entries such as place names and sometimes people. Because the format of the dateline is regular given a particular source for documents rules for parsing these datelines could be created. For this step, the TAC 2014 document collection was used as a proxy for the document types that would be seen during the task. This added 6,078 mentions to the output or 0.03%. Although this was a small percentage, many queries referenced entities in the headline in TAC2014, therefore no answer was possible for such a query. Similar analysis has not been done on the TAC2015 queries to understand the importance of headlines to performance.

3.3 Cross-Document Entity Coreference
In 2013 we developed a tool for cross-document coreference named Kripke that takes as input a serialized TAC knowledge base and produces equivalence sets that encode entity coreference relations. Our motivation for a new tool was that we wanted an easy-to-run, efficient, and precision-focused clusterer; previously (i.e., in 2012) we had used string-matching alone, or a Wikipedia-based entity linker.

Kripke is an unsupervised, procedural clusterer based on 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 generally sufficient for disambiguation. 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.6 Related indexing techniques are variously known as

6Support for orthographically dissimilar name variants (i.e., aliases) was planned, but not implemented in time for this year.
blocking (Whang et al., 2009) or canopies (McCal- 
lum et al., 2000).

Currently, 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 cut-
off, then the contextual similarity is deemed ade-
quate. Such a technique should be able to tease apart
George Bush (41st president) and his son (43rd pres-
ident) 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 clus-
tering passes, where in each subsequent pass con-
ditions 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. Additional details can be found in
(Finin et al., 2014).

3.4 KB Cleanup and Slot Value Consolidation

This step, which is repeated several times in the
pipeline ensures that all relations have their inverses
in the KB, culls relations that violate type or value
constraints, and reduces the number of values to
match expectations for each type of slot.

3.4.1 Inverses Relations

Producing inverses is an entirely deterministic pro-
cess 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 rela-
tions are often extracted in only one direction dur-
ing document-level analysis, yet we want both asser-
tions to be explicitly present in our KB to aid with
downstream reasoning.

Figure 1: Kelvin initially extracted 121 distinct
values for Barack Obama’s employer from 26,000
Washington Post articles. The number of attesting
documents for each followed a power law, with nine
documents for the most popular value only one for
the majority.

3.4.2 Predicate Constraints

Some assertions extracted from SERIF or FACETS
can be quickly vetted for plausibility. For exam-
ple, the object of a predicate expecting a coun-
try (e.g., per:countries_of_residence) must match
a small, enumerable list of country names; Mas-
sachusetts is not a reasonable response. 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 an accepted list of re-
sponses (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).

3.4.3 Consolidating Slot Values

Extracting values for slots is a noisy process and er-
rors are more likely for some slots than for others.
The likelihood of finding incorrect values also de-
pends on the popularity of both the entity and slot.
For example, in processing a collection of 26K ar-
ticles from the Washington Post, we observed more
than fifty entities who had 14 or more employers.
One entity was reported as having had 122 employ-
ers (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 docu-
ments to rank candidate values, with greater weight
given to values that were explicitly attested rather
than implicitly attested via inference rules. See Fig-
ure 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. Table 1 shows the thresholds we used for some predicates. For predicates in our table, we accepted the \( n \)th 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.

### 3.5 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 one is a grandparent of the other. Our knowledge of geographic subsumption produced a 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_or_province_of_birth is Maryland.

### 3.6 Linking to External Knowledge Bases

Entities are linked to one more external knowledge bases. Our current system uses just one external KBs, the version of the Freebase KB described in Section 4. Our approach is relatively simple, only comparing an entity’s type and mentions to the external KB’s entities types, names and aliases.

In linking a collection entity to a KB entity, we start by producing a candidate set by querying the KB all of its entities whose names or aliases match any of the collection entity’s canonical mentions\(^7\). The candidates are ranked by counting how often each matching mention was used and by the KB entity’s significance score. We used experimentally derived thresholds to reject all candidates if there were too many or the top score was too low relative to the second highest score.

### 3.7 Knowledge-Level Clustering

After analyzing our 2014 Cold Start performance, we identified that KELVIN often under-merged en-

\[^7\]Matching is done after normalizing strings by downcasing and removing punctuation.
ities. We added additional inference rules for merging entities that were applied at the knowledge-base level. One set of rules merges entities that are linked to the same Freebase entity. Another set merges entities that share the same canonical mention under several entity type specific conditions. For example, two ORG entities with sub-type Educational are merged if they have the same canonical mention and the mention includes a token implying they are higher-ed organizations (e.g., college, university or institute).

A third set merges entities based on “discriminating relations”. Our intuition is that it is likely that two people with similar names who have the same spouse or were born on the same date and in the same city should be merged. Similarly, organizations with similar names who share a top-level employee are good candidates for merging.

We maintain three categories of relations, those with high, medium and low discriminating power. Example of highly discriminating relations are per:children, org:date_founded and gpe:part_of. Medium discriminating relations include per:city_of_birth, gpe:headquarters_in_city, and org:member_of. Examples of relations with low discriminating power include per:stateorprovince_of_birth, org:students, and gpe:deaths_in_city. The decision to merge two entities with similar names is dependent on their type and the number of high, medium and low discriminating relations they share.

### 3.8 Selecting Provenance Metadata

This step selects the provenance strings to support each relation for the final submission. The 2015 evaluation rules allow for up to four provenance strings to support a relation, none of which can exceed 150 characters. For simple attested values, our initial provenance strings are spans selected from the sentence from which we extracted the relation, e.g., “Homer is 37 years old” for a per:age relation. Inferred relations can have more than one provenance string which can come from different documents, e.g., “His daughter Lisa attends Springfield Elementary” and “Maggie’s father is Homer Simpson” for a per:siblings relation.

An initial step is to minimize the length of any overly-long provenance strings to select a substring that spans both the subject and object. Candidate provenance strings whose length exceeds the maximum allowed after minimization are discarded. If there are multiple provenance candidates, a simple greedy bin packing algorithm is used to include as many as possible into the four slots available. Preference is given for attested values over inferred values and provenance sources with higher certainty over a those with lower.

### 3.9 Post-Processing

The final steps in our pipeline produces several outputs: a submission file that complies with the task guidelines and an RDF version that is can be loaded into a triple store for inspection and querying.

We start by normalizing temporal expressions, ensuring that all entities have mentions, insisting that relations are consistent with the types of their subjects and objects, confirming that logical inverses are asserted, and checking that entities have mentions in the provenance documents.

We developed Tac2Rdf to translate a KB in TAC format to RDF using an OWL ontology 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 (Rietveld and Hoekstra, 2013). We translated the 250 2014 Cold Start evaluation queries into SPARQL and found that this provided an easy way to test our system as we developed the current version.

### 4 Knowledge Base

We created a knowledge base derived from the BaseKB version of Freebase that was distributed by
the LDC for use in the 2015 TAC KBP EDL tasks. This was used to support both our Cold Start and TEDL submissions.

The full BaseKB dataset is quite large, containing more than a billion facts (counting each triple as a fact) about more than 40 million subjects. Much of this information is not relevant to the KBP tasks, such as information about musical groups, films or fictional characters.

We started by identifying entities that might be relevant to the TAC KBP tasks and removing any triples whose subjects were not in this set. An initial step was to identify those subjects that mapped to one of the five standard TAC types (PER, ORG, GPE, LOC and FAC) or represented what Freebase calls a Compound Value Type or CVT. One issue in doing this is that the TAC ontology assumes that its five types are disjoint, but relevant Freebase entities can have types that map to several TAC types. For example, the Freebase entity with canonical name Oval Office (m.01hhz7) has subtypes associated with both a LOC and an ORG. We used various heuristics to assign such entities to only one TAC type.

We kept information about any CVTs that were linked to a TAC relevant entity. CVTs are used in Freebase to represent reified relations, such as relations that have associated data, such as units (for measurements), time or location.

Triples with literal values (i.e., strings) for objects are tagged with an XSD data type (e.g., integer or date) or a language tag (e.g., @EN for English or @ZH for Chinese). We discarded any string values whose language tag was not in the English, Chinese or Spanish families.

We computed a measure of an entity’s significance based on the number on triples in which it was the subject or object. The significance was set as the base-2 log of the total number of links. For the reduced KB, this was a real number between 1 and 20. Table 2 shows data for a few example entities.

Finally, we added additional assertions to record an entity’s TAC type and normalized versions of an entity’s names and aliases by downcasing, removing punctuation, entity significance, number of in- and out-links, etc. The reduced KB has 146M triples about more than 4.5M TAC entities: 3074k PERs, 686k ORGs, 539k GPEs, 161k FACs and 85k LOCs. It was loaded into a triple store with SPARQL endpoint using the Apache Jena suite of RDF tools: Jena, Fuseki and TDB.

---

### Table 3: Ground-truth, right, wrong and duplicate answers for our submitted 2015 runs.

<table>
<thead>
<tr>
<th>Run</th>
<th>0-hop</th>
<th>1-hop</th>
<th>All-hop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GT</td>
<td>R</td>
<td>W</td>
</tr>
<tr>
<td>1</td>
<td>3935</td>
<td>606</td>
<td>693</td>
</tr>
<tr>
<td>2</td>
<td>3935</td>
<td>834</td>
<td>1081</td>
</tr>
<tr>
<td>3</td>
<td>3935</td>
<td>649</td>
<td>741</td>
</tr>
<tr>
<td>4</td>
<td>3935</td>
<td>849</td>
<td>1100</td>
</tr>
<tr>
<td>5</td>
<td>3935</td>
<td>853</td>
<td>1078</td>
</tr>
<tr>
<td>6</td>
<td>3935</td>
<td>652</td>
<td>1485</td>
</tr>
</tbody>
</table>

### Table 4: Micro precision, recall and F1 scores for our submitted 2015 runs.

<table>
<thead>
<tr>
<th>Run</th>
<th>0-hop</th>
<th>1-hop</th>
<th>All-hop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F1</td>
</tr>
<tr>
<td>1</td>
<td>0.4665</td>
<td>0.1540</td>
<td>0.2316</td>
</tr>
<tr>
<td>2</td>
<td>0.4355</td>
<td>0.2119</td>
<td>0.2851</td>
</tr>
<tr>
<td>3</td>
<td>0.4669</td>
<td>0.1649</td>
<td>0.2438</td>
</tr>
<tr>
<td>4</td>
<td>0.4356</td>
<td>0.2158</td>
<td>0.2886</td>
</tr>
<tr>
<td>5</td>
<td>0.4417</td>
<td>0.2168</td>
<td>0.2908</td>
</tr>
<tr>
<td>6</td>
<td>0.3051</td>
<td>0.1657</td>
<td>0.2148</td>
</tr>
</tbody>
</table>

---

The dataset is available from the Linguistic Data Consortium as LDC2015E42.
5 TEDL System Description

We submitted three runs that largely used the KELVIN system with some additional steps and two based on new components for named entity recognition and entity linking. Both used a version of the BaseKB Freebase dump distributed by the LDC for use in 2015 EDL and described in Section 4.

5.1 KELVIN-based runs

We submitted three runs that used the Kelvin’s components with some additional steps. The overall processing can be viewed as having three stages: monolingual document processing, multilingual cross-document co-reference resolution, and multilingual knowledge-base processing.

The first stage applied Kelvin’s standard pipeline to each of the three monolingual document collections using the appropriate Serif language model\(^1\). For each of the three, we use just two of the files produced, the serialized TAC KB produced by Kelvin’s document level processing and the co-reference relations produced by Kripke.

The second stage starts by creating a multilingual document level KB by concatenating the three monolingual KBs. If the mention-translation option is enabled, English translations of of Chinese and Spanish mentions are added. For 2015 we used the Bing translation service API. This combined, multilingual collection is then processed by Kripke to produce cross-document coreference relations.

The co-reference relations from each of the monolingual collections and from the combined collection are integrated using a simple algorithm to combine equivalence relations, yielding a single co-reference clustering file for the entire collection. The three monolingual document-level KBs are then combined (without any translated mentions) and the cross-document coreference relations used to generate the KB for subsequent KB-level processing by the rest of Kelvin’s pipeline.

The remaining processing, including linking, was performed by Kelvin’s pipeline with a few small additions. We added a special module to find and extract authors’ names of posts in Bolt documents. The 2015 TEDL guidelines required nominal mentions to be noun phrases head noun or nominal compound, but our document-level processing typically produced longer nominal mentions. We added a simple module to POS-tag nominal mention and reduce them, if possible, to their first sequence of consecutive tokens tagged as NN, NNS, NNP or NNPS. Examples of our adjusted nominal mentions are shown below.

- a former two-term Florida governor → Florida governor
- the most formidable fundraiser in the Republican field → fundraiser
- Republican Congressman from New York → Congressman
- the Greek minister of Productive Reconstruction, Environment and Energy → Greek minister

5.2 Experimental runs

We submitted two runs based on new components for named entity recognition and linking. These used a version of Kripke for clustering, but did not rely on any other KELVIN modules.

5.2.1 Named Entity Recognition

For our Golden Horse NER system we adapted the model described in (Peng and Dredze, 2015; Peng et al., 2015). It is a Conditional Random Field (CRF) model with a modified objective function. The idea is to use large amount of unlabeled data and jointly trained embeddings to improve the NER quality. The modified CRF objective function is as follows:

\[
L_s(\lambda, e_w) = \frac{1}{K} \sum_k \left[ \log \frac{1}{Z(x)^k} + \sum_j \lambda_j F_j(y^k, x^k, e_w) \right],
\]

where \(K\) is the number of instances, \(\lambda\) is the weight vector, \(x^k\) and \(y^k\) are the words and labels sequence for each instance, \(e_w\) is the embedding for a word/character/character-position representation \(w\), \(Z(x)^k\) is the normalization factor for each instance, and \(F_j(y^k, x^k, e_w) = \sum_{i=1}^n \sum_{\text{adj}} f_j(y^k_i, y^k_{i-1}, x^k_i, e_w, i)\) represents the feature function in which \(j\) denotes...
different feature templates and $i$ denotes the position index in a sentence. The new objective function takes characters’ embeddings into account when doing NER, and also actively modifies the embeddings for the characters during training.

We pre-trained character-positional embeddings on the Xinhua News Agency portion of Chinese Gigaword by the standard skip-gram language model objective (Mikolov et al., 2013):

$$\mathcal{L}_u(e_w) = \frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j}|w_t), \quad (1)$$

where

$$p(w_i|w_j) = \frac{\exp(e_{w_i}^T e_{w_j})}{\sum_{i'} \exp(e_{w_i'}^T e_{w_j})}.$$ 

and used the joint training schema described in Peng and Dredze (2015) to train the log-bilinear CRF model. The final model is trained on the data from Sighan 06 shared task on Chinese NER and TAC training data.

5.2.2 Entity Linking

Slinky (Benton et al., 2014) is an entity linking tool that implements a highly parallel message passing infrastructure using Akka (Wyatt, 2013) and adopts an SVM learning-to-rank approach for entity disambiguation. To make Slinky applicable to 2015 TAC TEDL task, several significant modifications were required.

First, since the task involves trilingual entity linking involving English, Chinese and Spanish, new models were trained for each of these languages. For each language, the ranking model is trained in the following procedure with the SVM rank objective and only linear kernels are considered. Queries are first partitioned into 60% train, 20% dev, and 20% test set. The slack parameter $C$ is tuned on the dev set, where $C \in \{b \times 10^c\}$, $b \in \{1, 5\}$ and $e \in \{-5, -4, -3, -2, -1, 0, 1, 2\}$. The best-performing setting is trained on train + dev and evaluated on the test set. The final model is trained on all the data.

Second, Slinky’s original Wikipedia-based knowledge base was replaced with data extracted from the knowledge base described in Section 4. The data included key information on each of the 4.5 million entities in our subset of Freebase: Freebase ID, TAC type, and all names, aliases and descriptions tagged as English, Chinese or Spanish.

Third, Slinky was modified to work with the Concrete (Ferraro et al., 2014) representation we use for document-level information extracted from text. It takes its input as a Concrete object representing the document and inserts its output in the object.

6 Cold Start Submissions and Results

<table>
<thead>
<tr>
<th>Name</th>
<th>Link</th>
<th>Canonical Mention</th>
<th>Pattern Matching</th>
<th>Open IE</th>
</tr>
</thead>
<tbody>
<tr>
<td>hltcoe1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hltcoe2</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hltcoe3</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hltcoe4</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hltcoe5</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>post-hoc</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Experimental variables for submitted runs.

We submitted the maximum of five experimental conditions that started with a simplistic baseline pipeline, and which added individually Freebase linking, and cumulatively refining of canonical mentions and pattern matching. Run 5 had the same configuration as Run 4. The difference is in the random ordering of entities processed by KELVIN. This shows the variation introduced by this nondeterminism. Table 5 summarizes the various conditions and Tables 3 and 4 give the key performance

<table>
<thead>
<tr>
<th>run</th>
<th>entities</th>
<th>PER</th>
<th>ORG</th>
<th>GPE</th>
<th>facts</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>284072</td>
<td>153580</td>
<td>93084</td>
<td>37408</td>
<td>209075</td>
</tr>
<tr>
<td>2</td>
<td>271672</td>
<td>148227</td>
<td>86802</td>
<td>36643</td>
<td>206422</td>
</tr>
<tr>
<td>3</td>
<td>274003</td>
<td>152433</td>
<td>92859</td>
<td>28711</td>
<td>204232</td>
</tr>
<tr>
<td>4</td>
<td>273989</td>
<td>152424</td>
<td>92853</td>
<td>28712</td>
<td>210508</td>
</tr>
<tr>
<td>5</td>
<td>273983</td>
<td>152405</td>
<td>92859</td>
<td>28719</td>
<td>210136</td>
</tr>
<tr>
<td>6</td>
<td>266579</td>
<td>144255</td>
<td>84711</td>
<td>23738</td>
<td>226101</td>
</tr>
</tbody>
</table>

Table 6: Number of entities mentions and facts identified in the evaluation corpus for each run.
metrics. Finally, Table 7 compares the number of assertions found by SERIF/FACETS compared with pattern matching and the open information extraction.

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. The number of reported entities is generally similar in each run, with differences likely attributable to changes in cross-document entity coreference. This is partly due to non-determinism, but mostly because both hltcoe2 and hltcoe3 introduced modifications that were specifically aimed at effecting cross-document coreference. This focus came from analysis of the 2014 results where we observed many errors due to under merging of entities across documents. Although Run 1 had the most entities, this problem likely needs more attention.

6.1 Discussion

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

It appears that improving the canonical mentions (hltcoe1 vs. hltcoe3) positively impacts precision, which has a big impact of the 1-hop queries. The intuition is that with greater accuracy at cross document coreference, there are fewer unrelated entities that get included when moving from 0-hop to the 1-hop.

Comparing hltcoe3 to hltcoe4 and hltcoe5, the difference is in the additional extraction method using pattern matching. This is a noisy process, so it is unsurprising that precision goes down while recall improves for the 0-hop and overall.

Finally, looking at the run with open information extraction, the post-hoc scoring reveals that the extractions were too imprecise to have a positive effect on the overall outcome.

7 TEDL submissions and results

We submitted five TEDL runs, three of which used KELVIN for most of the processing and two of which were based on independent components. None of the runs used links to Wikipedia in the reference, used relations encoded in the reference KB, or attempted to generate meaningful confidence val-

Table 7: This table shows the number of assertions for each slot that were asserted by SERIF/FACETS and inference from hltcoe1, added by pattern matching from hltcoe5, and open IE from the post-hoc scoring.
1 | pre | rec | F1 | pre | rec | F1 | pre | rec | F1  
---|-----|-----|----|-----|-----|----|-----|-----|----
1 | .66 | .64 | .65 | .55 | .53 | .54 | .56 | .54 | .55  
2 | .66 | .64 | .65 | .55 | .53 | .54 | .56 | .54 | .55  
3 | .66 | .64 | .65 | .54 | .52 | .53 | .55 | .53 | .54  
4 | .63 | .22 | .33 | .40 | .14 | .21 | .53 | .18 | .27  
5 | .61 | .21 | .31 | .40 | .14 | .21 | .51 | .18 | .27  

Table 8: This table shows the precision, recall and F1 scores over all three languages for each run for three key metrics: strong typed mention match, strong all match and mention ceaf.

<table>
<thead>
<tr>
<th>Language</th>
<th>Submissions</th>
<th>NER</th>
<th>Linking</th>
<th>Clustering</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>6</td>
<td>3rd</td>
<td>3rd</td>
<td>4th</td>
</tr>
<tr>
<td>eng</td>
<td>8</td>
<td>3rd</td>
<td>6th</td>
<td>4th</td>
</tr>
<tr>
<td>cmn</td>
<td>7</td>
<td>2nd</td>
<td>2nd</td>
<td>2nd</td>
</tr>
<tr>
<td>spa</td>
<td>7</td>
<td>6th</td>
<td>6th</td>
<td>6th</td>
</tr>
</tbody>
</table>

Table 9: This table shows our relative ranking for our best run for the three key metrics: strong typed mention match, strong all match and mention ceaf.

7.1 Discussion

Our document-level processing components do a good job on in-document coreference detection. We experimented with several variations on entity clustering for multi-lingual collections. We used Kelvin on each monolingual collection separately. We then applied our Kripke agglomerative entity clustering system over the results and followed this with additional KB-level inference and clustering and linking.

Kripke produces co-reference relations from a TAC KB which are then used to merge entities and produce a new KB. We experimented with running Kripke on various KB combinations and combining the co-reference relations. We got the best results by integrating the co-reference relations obtained from each monolingual collections and those from the trilingual collection augmented with additional English mention strings produced by translating Chinese and Spanish mentions.

Table 10 data from our best TEDL evaluation run showing the number of entities before and after clustering, the number of (non-singleton) clusters, and the percent reduction in the number of entities for the monolingual collections and for five different methods on the multilingual collection. In describing the methods, \( K(x) \) is the result of applying Kripke to the monolingual collection \( x \), \( K(e + c + s) \) is the result of applying Kripke to the trilingual collection, \( X(e + c + s) \) is the KB that is the trilingual collection augmented with English translations of Chinese and Spanish mentions, and adding the results of two Kripke applications means integrating their coreference relations.

The final int2 approach yields the smallest number of entities. Showing that it is an actual improvement requires scoring complete runs based on the five options, which we have not yet done. However, we did experiment with the options on the training data and found the final one did produce the best score. In general, we find that the number of distinct entities is further reduced by about another 10% after applying inferencing and linking and that this also improved our score on the training data. For this evaluation run, we ended up with 5310 entities, a 54% reduction in the initial number of document-level entities.
8 Conclusion

The JHU Human Language Technology Center of Excellence has participated in the TAC Knowledge Base Population exercise since its inception in 2009, in Cold Start task since 2012, and in Trilingual Entity Linking and Discovery beginning this year. We modified the KELVIN system used in the 2012, 2013, and 2014 Cold Start task by introducing a Freebase based linking system and additional relation extractions for the Cold Start task. We further extended the KELVIN system for the Trilingual Entity Linking and Discovery task by adding support for documents in Chinese and Spanish, improving nominal mentions, developing new techniques for cross-lingual co-reference, and implementing a module for linking entities to Freebase.

Table 10: This table shows the effectiveness of strategies for combining monolingual and trilingual clustering data.

<table>
<thead>
<tr>
<th></th>
<th>before</th>
<th>after</th>
<th>clusters</th>
<th>reduction</th>
<th>method</th>
</tr>
</thead>
<tbody>
<tr>
<td>eng</td>
<td>4966</td>
<td>3131</td>
<td>567</td>
<td>40%</td>
<td>K(e)</td>
</tr>
<tr>
<td>cmn</td>
<td>3303</td>
<td>1858</td>
<td>404</td>
<td>46%</td>
<td>K(c)</td>
</tr>
<tr>
<td>spa</td>
<td>3161</td>
<td>1906</td>
<td>357</td>
<td>40%</td>
<td>K(s)</td>
</tr>
<tr>
<td>cat1</td>
<td>11430</td>
<td>6816</td>
<td>1328</td>
<td>40%</td>
<td>K(e)+K(c)+K(s)</td>
</tr>
<tr>
<td>cat2</td>
<td>11430</td>
<td>6617</td>
<td>1283</td>
<td>42%</td>
<td>K(e+c+s)</td>
</tr>
<tr>
<td>int1</td>
<td>11430</td>
<td>6475</td>
<td>1176</td>
<td>43%</td>
<td>K(e+c+s)+K(e)+K(c)+K(s)</td>
</tr>
<tr>
<td>tran</td>
<td>11430</td>
<td>6125</td>
<td>1205</td>
<td>46%</td>
<td>K(X(e+c+s))</td>
</tr>
<tr>
<td>int2</td>
<td>11430</td>
<td>5943</td>
<td>1090</td>
<td>48%</td>
<td>K(X(e+c+s))+K(e)+K(c)+K(s)</td>
</tr>
</tbody>
</table>

References

Adrian Benton, Jay Deyoung, Adam Teichert, Mark Dredze, Benjamin Van Durme, Stephen Mayhew, and Max Thomas. 2014. Faster (and better) entity linking with cascades. In *NIPS Workshop on Automated Knowledge Base Construction*.


Francis Ferraro, Max Thomas, Matt Gormley, Travis Wolfe, Craig Harman, and Benjamin Van Durme. 2014. Concretely annotated corpora. In *4th Workshop on Automated Knowledge Base Construction (AKBC)*.


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*, Gaithersburg, MD, Nov.


Nanyun Peng, Francis Ferraro, Mo Yu, Nicholas Andrews, Jay DeYoung, Max Thomas, Matt Gormley, Travis Wolfe, Craig Harman, Benjamin Van Durme, and Mark Dredze. 2015. A chinese concrete nlp pipeline. In *North American Chapter of the Association for Computational Linguistics (NAACL), Demonstration Session*.


Derek Wyatt. 2013. *Akka Concurrency*. Artima Incorporation, USA.