HLTCOE Participation at TAC 2012: Entity Linking and Cold Start Knowledge Base Construction

Paul McNamee, Veselin Stoyanov, James Mayfield
Johns Hopkins University
Human Language Technology Center of Excellence

Tim Finin, Tim Oates
University of Maryland
Baltimore County

Tan Xu, Douglas W. Oard
University of Maryland
College Park

Dawn Lawrie
Loyola University Maryland

Abstract

Our team from 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 entity linking evaluations in 2009, 2010, and 2011. This year we developed two new systems: CALE (Context Aware Linker of Entities) and KELVIN (Knowledge Extraction, Linking, Validation, and INference) to support our research for this year's exciting tasks.

1 Introduction

The JHU HLTCOE has participated in the TAC Knowledge Base Population exercise since its inception in 2009. Our focus on entity linking over the past year was to develop a new, context-oriented entity linking system, CALE, which explicitly leverages contextual cues when resolving entity mentions. The new system, like our previous systems, uses phases for candidate selection and ranking; however, unlike our 2010 and 2011 systems, this year we exploited large external resources that aid resolution to Wikipedia-derived KBs. Additionally, CALE applies novel joint inference algorithms to achieve better global resolution of all entity mentions in a document.

This year in the first running of the Cold Start task, we also developed another new system, KELVIN, which creates knowledge bases in accordance with the Cold Start task specifications. We used multiple layers of NLP software in our approach. The most significant piece was the BBN SERIF tool, a NIST ACE entity/relation/event detection system. We also used a maximum entropy model for extracting personal attributes (FACETS), the CUNY KBP Slot Filling Toolkit, the HLTCOE CALE system for entity linking, and various components for lightweight inference, time normalization, fusion of evidence, and sanity checking. We found the task both complex and interesting.

In the rest of the paper we describe both systems, our results on the TAC-KBP tasks, and some preliminary analysis.

2 Context Aware Linking of Entities

We have been participating in the entity linking task since its inception in 2009 with several versions of our research code stitched together with scripts. One of the main goals for this year was to develop a platform for entity linking that will support all of our entity linking needs. Those needs include monolingual and cross-lingual entity linking, entity linking inside the KELVIN system, and ongoing efforts in other domains such as first person communication (*e.g.,* email) and informal communication (*e.g.,* Twitter messages). The new platform that we developed this year is named CALE for Context Aware Linker of Entities. CALE was conceived and developed as a general entity linking architecture that can be applied to many tasks. We instantiated and used it this year for the general TAC entity linking task as well as the Spanish cross-lingual entity linking.

CALE follows the general architecture of our system from last year (McNamee et al., 2011). Like many other successful entity linking systems, CALE utilizes two phases: *triage*, during which we collect a list of viable candidate entities for each mention, and *ranking*, during which we rank the entities in the candidate list and select the highest scoring entity. Unlike our previous system, CALE was designed to incorporate the notion of context as described in Stoyanov et al. (2012). Unfortunately, most of the context-based technology that we are developing proved immature, so the runs that we submitted do not utilize the iterative context refinement procedure described in Stoyanov et al. (2012).

Nevertheless, CALE incorporates novel machine learning techniques, which allow it to be more accurate than its predecessor as empirical results suggest. We implemented three recent advances in our system: we take advantage of document context by using joint inference; we utilize structure prediction cascades; and we apply loss- and approximation-aware training through the Empirical Risk Minimization under Approximations (ERMA) algorithm. These advances are described in more detail below:

Joint Prediction. We identified the utilization of context as the single most significant area for improvement of our entity linking system. For instance, if a single document contains the mentions "Dublin" and "CA", it will be likely that the former refers to "Dublin, California" and the latter to "California," the US state. To utilize context, CALE applies the following steps:

1. Finds the place in the document where the query mention is located. This step is necessary because traditionally TAC queries have not included offset information.

2. Adds all Named Entities (NEs) that are in the same paragraph as the mention. Most of the TAC queries contain a single mention per document, so we need to add mentions on our own.

3. Defines a Markov Random Field (MRF) over the resolution decisions. The MRF contains one random variable (RV) for each mention. The value of the RV corresponds to the entity to which the mention is resolved. The domain of each RV is the list of possible candidates as determined by the triage phase. The MRF also contains a unary factor connected to each random variable. All traditional ranking features are defined over this factor. We also include one binary factor connecting each pair of random variables. These factors consist of pairwise features that capture the compatibility of particular pairs of resolutions. For instance, we have a feature that fires if the two entities in the particular resolution configuration link to each other (using Wikipedia links). The hope is that we will learn a high weight for this feature, which will give advantage to joint resolutions that are linked in Wikipedia. This feature will fire, for instance, in the above example when "Dublin" is resolved to "Dublin, CA" and "CA" is resolved to California since there is a direct link in Wikipedia between the two.

4. During testing, CALE runs inference over the MRF to determine the joint configuration that maximizes the model score. Since in most cases exact inference is intractable (when we have more than 3 mentions, it is infeasible to enumerate all possibilities), we use loopy belief propagation to approximately find the most likely joint configuration. We resolve each mention to its assignment under that configuration.

During training, we perform the same steps, except that the final step consists of training classifier weights instead of resolution. By default, we can train the MRF feature weights using maximum likelihood estimation. This procedure is approximate since it requires inference, which is approximate in most cases.

Structured Prediction Cascades. Using the architecture described above, CALE can perform joint inference, which allows it to account for context in which each mention is resolved. In terms of runtime, the extra cost of the approximate joint inference is due to the pairwise factors between each pair of random variables $\{v_i, v_j\}_{i \neq j}$. Each such factor has $n_i \cdot n_j$ entries, where n_i stands for the number of candidate resolutions for variable v_i. Although triage limits the number of candidates for each mention, CALE still considers an average of 90 candidates per mention. Thus, for a typical case, a factor may have 90^2=8100 entries in its conditional table and in some cases this number may be much higher. There are $O(n^2)$ pairwise factors for a document with n mentions. Therefore, using joint inference over all candidates proved impractical for all but the smallest examples. For that reason, we need to additionally limit the number of candidates that we consider for joint inference. One way to limit the number of candidates is through the use of structured prediction cascades.

Structured prediction cascades (Weiss and Taskar, 2010) represent a technique for structured classification which relies on a sequence of models that increase in complexity while progressively pruning the space of possible outputs. In our case, the cascade consists of two such classifiers – a traditional classifier that relies on unary features and the joint classifier that adds pairwise binary features. We use the former to limit the number of candidates that we consider (on a typical run we reduce from an average of 90 to an average of 3 candidates). Once the candidates are trimmed down, we incorporate the pairwise features that allow us to capture effects of context. Note that an effective cascade requires that the initial classifiers are trained for a special loss function that encourages aggressive trimming, but emphasizes the cost of dropping the correct answer.

ERMA. Finally, we utilize the ERMA training algorithm (Stoyanov et al., 2011; Stoyanov and Eisner, 2012) to train the MRF parameters through empirical risk minimization as opposed to the traditional maximum likelihood training. The ERMA algorithm allows us to learn a classifier that is aware both of the test-time loss as well as the test-time approximations with which the system will be used. Loss-aware training is particularly important for the specialized loss function needed by the first stage of the structured prediction cascade. Filtering loss (Weiss and Taskar, 2010) aims to reward classifiers that remove a large percentage of the candidates, while including a large penalty for removing the correct answer from the list. The classifier for the second stage of the cascade is optimized by maximizing training accuracy, which is a reasonable surrogate for the BCubed+ score used for evaluation.

Experiments. Prior to the TAC evaluation run, we performed experiments to evaluate the contributions of the novel learning techniques and gauge the overall CALE performance. We tested on the 2011 TAC KBP test set, while training on all documents from TAC KBP 2009 and 2010. Table 1 summarizes the accuracy and BCubed+ scores for our experiments. Compared to our old system (McNamee, 2010), listed in the first row of the table, CALE exhibits superior performance even before adding any of the new machine learning components. We attribute this to the streamlining of the system and the addition of some new features. This year we are also utilizing a new resource: the Google dataset (Spitkovsky and Chang, 2012) consisting of the anchor text of links to Wikipedia articles. We use the Google data both in the triage phase (by adding all entities to which links with the particular anchor text point) and as a feature in the ranking step.

	Accuracy	Bcubed+
Old System	0.775	0.74
CALE	0.8401	0.815
+ Cascades	0.8284	0.801
+ ERMA	0.8471	0.819
+ Cascades and ERMA	**0.8631**	**0.832**
+ Joint Features	0.8586	0.828

Table 1: Entity Linking results on 2011 TAC KBP.

Compared to the baseline CALE implementation (the second row) adding structured prediction cascades (the third row) leads to a decrease in performance, while adding loss-sensitive training through ERMA improves performance only slightly. Incorporating both structured prediction cascades and ERMA training leads to a sizable improvement in performance (over 2 points in accuracy and 1.7 points of BCubed+ score). Note that this setting still does not use the joint futures: we use a two stage cascade approach, but the second stage of the cascade relies on the same unary features as the first stage. Nevertheless, the system performance improves consistently – we tried several testing sets and confirmed the effect. We attribute the improvement on the fact that the second stage of the cascade, while relying on the same features, can now focus on the most probable answers only.

Adding the joint features did not lead to an improved performance to our disappointment. In experiments following the TAC evaluation we discovered a major bug to our joint feature computation, which lead to features being assigned to the wrong entries configurations of entities. Because we did not observe any improvement, our Entity Linking runs did not rely on the joint features but did utilize the structure prediction cascade and ERMA training.

2.1 Spanish Entity Linking

For the Spanish entity linking task we utilized a translation system–Joshua (Ganitkevitch et al., 2012) to translate Spanish documents into English. We utilized a translation model trained on Europarl (Koehn, 2005) and News Commentary Spanish-English parallel corpora[1] and tuned against News Commentary.

Our Spanish system was based on (English) CALE with modifications kept to a minimum. We experimented with two different processing strategies:

1. **SEL1** translates documents into English and uses CALE unmodified.

2. **SEL2:** modifies CALE components that rely on document text so that they would work cross-lingually.

The implementation of SEL1 is quite straightforward, while for SEL2, we had to modify several components. The triage phase of SEL2 was unmodified since: i) Spanish uses Roman script – we can use the same name look-up strategy as the English system after stripping diacritics and ii) the aforementioned Google dataset (Spitkovsky and Chang, 2012) is multilingual, so we can use it without modification. For the ranking step, SEL2 simply substituted translated text for original text in the features that capture text comparability. Additionally, we concluded that both for triage and ranking it is very important to recognize the full set of named entity (NE) mentions in the original text. We utilized the named entity mentions to expand the single-name queries (*e.g.,* "Rodriguez") to the longest NE containing the name (*e.g.,* "Alex Rodriguez"). This name expansion pre-processing step resulted in about 10 points improvement in performance on the KBP 2012 Spanish training queries. For named entity recognition (NER), we trained our own system, as described below.

	Best	HLTCOE		Median
		Official	Corrected	
English	0.730	0.699	**0.709**	0.536
no text	0.730	**0.660**	**0.660**	0.522
Spanish	0.643	0.632	**0.641**	N/A

Table 2: Summary of the B^3+ scores on the 2012 TAC KBP task. Our best results are shown in bold.

Spanish NER For this task, TnT (Brants, 2000) was used train on CoNLL 2002 training data. Accuracy on testa approached 95%. For the CALE system, TnT was trained on a combination of training, testa and testb data, since these test sets do not overlap with the documents used in the Spanish queries and slightly increasing the training size gives a slight improvement to performance.

2.2 Results

Table 2 summarizes our entity linking results. Note that the table contains two columns for our best results. Due to an unfortunate error, for both the English and the Spanish task we submitted the wrong file for the second submitted run. In the case of English task we submitted the third run twice – as the second and the third run. In the case of the Spanish task, we submitted the first run twice. Upon receiving the results we realized our mistake and scored the results of the previously run and saved results. In both cases our second run scored the best. We are including these results in this paper for completeness although they are not part of our official scores.

Our scores indicate that our system scored closely behind the top scoring systems on both the English and the Spanish tasks. We consider this to be a success since we used a general architecture with little customization to the TAC task.

Detail results for all runs that we submitted are shown in Table 3. Those results include the corrected scores for the second English and the second Spanish runs. Below is a description of all the runs that we submitted:

English.

1. **EngRun1:** Our default English run. It relies on the standard triage and ranking approach and utilizes the Google data set. Also uses

[1] http://statmt.org/wmt12/translation-task.html

structured prediction cascades and trains using ERMA. Following the ranking phase, NIL resolution is performed by using exact string match.

2. **EngRun2:** Same as above, except for NIL match. In this version we maintain a separate database of added entities and use a resolution approach identical to the ranking approach used in the base system to perform NIL resolution.

3. **EngRun3:** Same as 1., with the addition of features that indicate the presence of particular words in the entity title. The words that we target are common nouns such as "album" or "river" that sometimes occur in entity names. We compiled a list of such words by picking out the most frequent common words in Wikipedia titles.

4. **EngRun4:** The default system (1.) tries to expand the query strings against the NEs extracted from the document before resolving. This version does not.

5. **EngRun5:** This version does not use the text of the Wikipedia articles (i.e., excludes the features that rely on article text). Everything else is the same as **EngRun1**.

Spanish. As previously mentioned, we experimented with two strategies: SEL1 translates the Spanish documents into English and uses a retrained version of the monoligual CALE system; SEL2 modifies CALE to process Spanish queries crosslingualy.

In this year's Spanish cross-lingual entity linking evaluation, 75 out of 2066 queries come from English documents. The SEL1 strategy process those queries together with the translated Spanish documents, while SEL2 process them separately using the trained English monolingual system.

Here is a summary for the five Spanish runs that we submitted:

1. **SpaRun1:** This run applies strategy SEL2. English queries, are resolved with the **EngRun1** system. Spanish queries are first expanded to the longest NE that contains the mention string using the NEs found by the Spanish NE tagger. The rest of this run is similar to **EngRun1**, except for using the translated document for the features that require document text. The run utilizes the Google data, uses structured prediction cascades and trains using ERMA. NIL resolution is performed by using exact string match.

2. **SpaRun2:** This run is identical to **SpaRun1** except for NIL resolution. Instead of exact string match, this runs uses the ranking approach described before.

3. **SpaRun3:** This run applies strategy SEL1: translates Spanish documents into English, then the run is identical to **EngRun1**.

4. **SpaRun4:** Same as **SpaRun3** except for NIL resolution. While **SpaRun3** uses exact string match, this run uses the ranking approach.

5. **SpaRun5:** Same as **SpaRun3**, but trained on a larger dataset. After translation, the SEL1 strategy treats Spanish queries the same way as English. Thus, we can utilize monolingual English queries to enlarge the training set. In this run, we add the queries from the TAC KBP 2011's English entity linking task to the training queries for this year's Spanish task.

3 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 which we used was a NIST ACE entity/relation/event detection system, the BBN SERIF system. In addition to SERIF, we relied on: a maximum entropy trained model for extracting personal attributes (FACETS, another BBN tool); a KBP Slot Filling system (CUNY KBP Toolkit); entity linking (the COE CALE system); and, components for lightweight inference, time normalization, fusion of evidence, and sanity checking.

3.1 Submitted Runs

We submitted 5 experimental conditions that started with a baseline pipeline, and which used (or didn't

	Accuracy	B^3+					
		All Mentions	InKB	NotInKB	PER	ORG	GPE
EngRun1	**0.755**	0.699	0.653	0.749	0.840	0.615	0.579
EngRun2	**0.755**	**0.709**	0.653	**0.771**	**0.854**	**0.628**	0.580
EngRun3	0.753	0.697	**0.655**	0.744	0.833	0.612	**0.584**
EngRun4	0.748	0.693	**0.655**	0.735	0.826	0.619	0.575
EngRun5	0.717	0.660	0.605	0.722	0.810	0.584	0.518
SpaRun1	0.714	0.482	0.538	0.394	0.296	0.552	0.510
SpaRun2	0.714	**0.641**	**0.540**	0.721	**0.801**	**0.624**	0.525
SpaRun3	0.701	0.520	0.489	0.525	0.511	0.524	0.504
SpaRun4	0.702	0.632	0.491	**0.745**	0.784	0.602	0.530
SpaRun5	**0.726**	0.543	0.528	0.533	0.524	0.539	**0.536**

Table 3: Results on the 2012 TAC KBP task (B^3+ F1).

Name	CUNY Toolkit	Entity Linking
hltcoe1	No	Normalized String Match
hltcoe2	Yes[3]	Normalized String Match
hltcoe3	No	CALE
hltcoe4	Yes	CALE
hltcoe5	Yes	Normalized String Match

Table 4: Description of conditions for five HLTCOE submitted Cold Start runs.

use) the CUNY Slot Filler and the CALE entity linker. Table 4 summarizes the various conditions.

None of our runs made any direct use of external resources such as live use of the Internet or commercial Web search, access to Wikipedia, or knowledge sources such as RDF triple stores or DBpedia.[2]

3.2 Pipeline Components

3.2.1 SERIF

BBN's SERIF tool[4] (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,[5] and include:

- identifying named-entities and classifying them by type and subtype;
- performing intra-document co-reference analysis, including named mentions, as well as co-referential nominal and pronominal mentions;
- parsing sentences and extracting intra-sentential relations between entities; and,
- detecting certain types of events;

Our pipeline for a text document starts with applying the baseline SERIF system to it producing an XML document. We also then run the FACETS module, described below, which adds additional annotations to the SERIF output. For each entity with at least one name 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.

SERIF discovers *ACE relations* between entities, so one important task is to map these, when possible, to corresponding TAC slots.

Figure 1 shows some of the approximately 50 rules that are used, where the fist column denotes a discovered ACE relation, the second and third specify the type and subtype of the relation argument entities (with a ? representing any) and the third the corresponding TAC relation (with nil representing no such relation). In some cases, the results represented heuristics in that we knew that the result was

[2] Clearly some of our components use linguistics resources such as parsers or supervised NER modules based on annotated corpora. One of our entity linkers uses the TAC-KBP KB (LDC2009E58[A-C]) to resolve mentions. We did not use Internet resources to validate posited slot fills, or even to suggest correct responses, which was done by some systems in previous TREC/TAC Question Answering evaluations.

[4] Statistical Entity & Relation Information Finding

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

GEN-AFF.Citizen-Resident-Religion-Ethnicity	PER.?	GPE.Nation	per:countries_of_residence
GEN-AFF.Citizen-Resident-Religion-Ethnicity	PER.?	GPE.County-or-District	nil
ORG-AFF.Employment	PER.?	GPE.?	per:employee_of
ORG-AFF.Employment	PER.?	ORG.?	per:employee_of
ORG-AFF.Student-Alum	PER.Individual	ORG.Educational	per:schools_attended

Figure 1: Examples of simple rules for mapping ACE relations into TAC relations

not always sound, but through experimentation we believed that it typically gave a good answer.

SERIF also discovers ACE events from which we can sometimes extract TAC slots. The mapping process was less regular for events, so each possible ACE event type was handled by a procedure which attempted to extract TAC slot data. For example, the ACE events *Life.Be-Born* and *Life.Die* were processed to identity the entity involved as well as the time and location (Country, city and district).

Although SERIF is very effective at finding mentions for named entities, we had to deal with several issues: overly long mentions, nested mentions and choosing a canonical mention. We noted that sometimes SERIF mistakenly took a phrase after a mention to be part of the mention, for example, identifying *" Mr. Powell, who may be considering a run for the White House"*. We used a simple technique to suppress these: rejecting any mention that whose number of words exceeded a threshold. The thresholds depended on the entity's type, *e.g.,* PER:7, ORG:10, GPE:8, and FAC:10.[6] SERIF produces nested mentions in some cases, even though this was not allowable in the ACE specification. For example Serif finds two entities in the string *"Baltimore, Maryland"*, the GPE.city *"Baltimore, Maryland"* and the GPE.state *"Maryland"*. Rather than untangling such nested mentions, we just used the outermost one. Finally, SERIF did not distinguish a canonical mention for each mention chain. We considered two heuristics: choosing the longest mention or choosing the one found earliest in the document. Using a large newswire development collection, we found that for about 97% of the entities, the longest and first mention in a chain were the same. We ended up selecting the longest mention for an entity as its canonical mention.

We extracted some additional custom non-TAC slots for entities which were passed on for use in subsequent processing. For example, for PER entities, we used the gender of any pronominal mentions to predict the entity's sex. We also recorded various ACE-related relation slots, such as generic family and business relations between two PER entities.

In Table 5 we list the most common slots that SERIF extracts from a set of Washington Post articles.

3.2.2 NER

We did not use the NIST-provided set of named-entities for the task. Instead we relied on SERIF's built-in detection of named persons, organizations, and GPEs.

3.2.3 FACETS

FACETS is an add-on package that takes SERIF input 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 role-specific attributes, such as employer for someone who has a job, (medical) specialty for physicians, or (academic) institution for someone associated with an educational institution.

In Table 6 we report the most prevalent slots extracted by FACETS from a collection of 26k Washington Post articles.

3.2.4 CUNY toolkit

In order to accomplish the slot filling task, the KBP Toolkit (Chen et al., 2011) developed at the CUNY BLENDER Lab was integrated into KELVIN. Given that the KBP toolkit was designed for the traditional slot filling task at TAC, the integration primarily involved creating the queries that the tool expected as input and parallelizing the toolkit to handle the vast

[6] Although facilities (ACE type FAC) were not generally relevant for TAC, we observed that many entities identified as FACs could also be reasonably interpreted as GPEs. Some of our ACE to TAC relation rules were thus triggered by FACs.

number of queries that are issued in the cold start scenarios.

A modified version of the SERIF output was used as the source information for the queries. A query is comprised of a query id, a mention from the document, the document id, an entity type, and a node id. Most of this information is straightforward to extract. The query id was generated from IDs assigned to the co-reference chains identified by SERIF. The document id was also readily available and all nodes where set to NIL,[7] given the nature of the Cold Start Task. The remaining information includes the mention and type of entity. Prior to running the KBP toolkit, canonical mentions for each entity are identified in the SERIF output. This canonical mention is used at the mention string for the query. In addition, SERIF assigns an entity type to each co-reference chain. This information is used for the entity type.

Thousands of queries are produced using this method: one per unique entity in each document. The corpus is then prepared for the KBP Toolkit and the output is integrated into KELVIN's results. More specifically, the pipeline of the KBP Toolkit first performs query expansion, followed by pattern matching, and finally, answer filtering (Chen et al., 2010). This pipeline follows a bottom-up form. Given an independent query, it first extracts all possible attributes to fill in pre-defined slots, and then for each within-query slot, it filters and merges answers. Lastly, it merges cross-query slots based on the query mention names and slot answers. Because it executes each query independently, we can parallelize the processing of queries from SERIF in addition to the cross-query slot merging step.

To gauge accuracy of extracted slots, some rough assessment was done over a collection of 807 New York Times articles that include the string "University of Kansas." From this collection, 4264 slots were identified. Nine different types of slots were filled in order of frequency: per:title (37%), per:employee_of (23%), per:cities_of_residence (17%), per:stateorprovinces_of_residence (6%), org:top_members/employees (6%), org:member_of (6%), per:countries_of_residence (2%), per:spouse (2%), and per:member_of (1%). For each of these nine types, ten slot-fills were randomly selected for evaluation to gain a rough approximation of the accuracy of the type. The accuracy ranged from 20% to 70%, with per:stateorprovinces_of_residence being most accurate. For instance, the toolkit deduced that Xiangdong Ji was a resident of Maryland from the statement "Dr. Ji of Maryland." However, it mistakenly asserts that Howard Dean is a resident of Missouri because he is mentioned in a sentence with Richard Gephardt, who is identified as being from Missouri. The worst performing slot was per:spouse. The poor performance in this case was caused by long sentences found in obituaries.

It should be noted that the system has a difficult time with obituaries in general where there tend to be long lists of people's names sprinkled with places of residence and their relationship to the deceased, such as "Husband of the late Katherine E. Weimer, he is survived by three daughters: Katherine Lasslob of Chalfont, PA, Barbara J. Blackwell and Patricia W. Hess, both of Princeton, eight grandchildren and his sister, Wilodean Rakestraw of Rochester, IN." In this case, Katherine Lasslob is identified as the wife of Paul K. Weimer rather than Katherine E. Weimer. The accuracy on non-obituaries was much higher.

When looking at the slot filling data from the KBP Toolkit for the Cold Start data, 17,941 slots were recognized. Figure 2 shows the distribution over the different slots. From this it can be seen that the distribution over slots is different from the newswire data.

3.2.5 Coreference

We used to methods for entity coreference. Under the theory that name ambiguity may not be a huge problem, we adopted a baseline approach of merging entities across different documents if their canonical mentions were an exact string match after some basic normalizations, such as removing punctuation and conversion to lower-case characters.

We also used the CALE system, described above, which links entities to the TAC-KBP KB. For entities that are not found in the KB, we reverted to exact string match. As can seen in our official results, CALE entity linking was the more effective approach for the Cold Start task.

[7] Meaning discover all learnable slots.

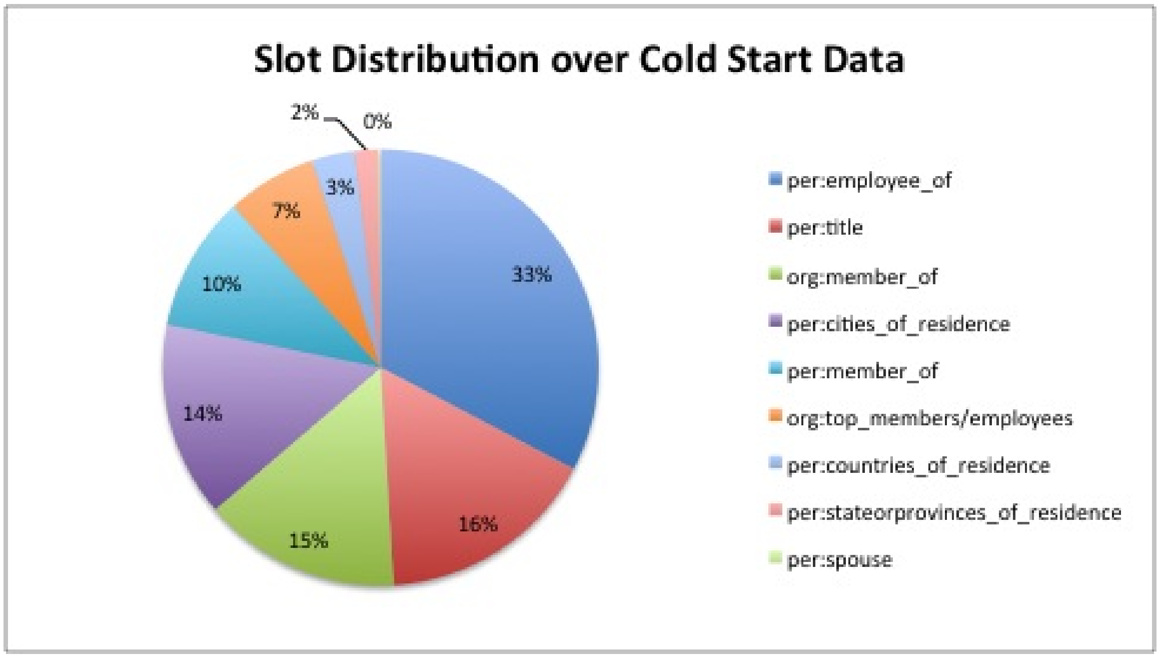

Figure 2: Distribution of slots in the Cold Start dataset learned using the CUNY slot filling toolkit

3.2.6 Timex2 Normalization

SERIF recognizes, but does not normalize, time expressions. We therefore used the Stanford SUTime package, a part of the Stanford CoreNLP toolkit to normalize all values in four fields: per:date_of_birth, per:date_of_death, org:date_founded, and org:date_dissolved. For relative references, we extract the document date from the second block of 8 characters in the file name (which conveniently happens to be in TIMEX2 format by convention) and pass it to SUTime as the reference date for every KB assertion found in that document.

3.2.7 Lightweight Inference

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

3.3 Development

As no suitable gold-standard KBs were available to us to assist during the development of the KELVIN system, we relied on qualitative assessment to gauge the effectiveness of the 40+ relations that can be asserted (and their inverses). We guess-timated that most relations were between 30-80% accurate. We created a variety of text collections on which to test the accuracy and scalability of our system. This included a 26k document collection of 2010 Washington Post articles from English Gigaword (5th ed.) which we hoped would be representative of the TAC KBP Cold Start evaluation collection. (We suspect it was not.)

KELVIN learns some interesting facts from the Washington Post articles:[8]

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

KELVIN also learns some true facts:

- Jared Fogle is an employee of Subway

[8] All 2010 Washington Post articles from English Gigaword 5th ed. (LDC2011T07)

Slotname	SERIF count
per:employee_of	60690
org:employees	44663
gpe:employees	16027
per:member_of	14613
org:membership	14613
org:city_of_headquarters	12598
gpe:headquarters_in_city	12598
org:parents	6526
org:country_of_headquarters	4503
gpe:headquarters_in_country	4503
org:subsidiaries	4222
per:cities_of_residence	3880
gpe:residents_of_city	3880
per:countries_of_residence	2881
gpe:residents_of_country	2881

Table 5: Top 15 prevalent slots extracted by SERIF from Washington Post test collection.

Slotname	FACETS count
per:title	44896
per:employee_of	39101
per:member_of	20735
per:countries_of_residence	8192
per:origin	4187
per:statesorprovinces_of_residence	3376
per:cities_of_residence	3376
per:country_of_birth	1577
per:age	1233
per:spouse	1057
per:parents	742
per:children	466
per:siblings	449
per:other_family	196
per:religion	190

Table 6: Top 15 prevalent slots extracted by FACETS from Washington Post test collection.

- 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

- Ian Soboroff is an employee of NIST[9]

- Ian Soboroff has per:title of computer scientist[10]

3.4 TAC-KBP Experiments

Our scores reported by NIST for the Cold Start task are given below in Table 7.

Use of CALE entity linking (*hltcoe3, hltcoe4*), which links more entities together than does exact string match, led to sizable improvements. Our best run (*hltcoe4*) made use of both CALE for entity linking and the addition of the CUNY slot filler.

4 Other Things

We built a simple query engine that enables searching a constructed KB. We also rendered the KB as a set of static HTML pages, which allows for exploratory browsing of the KB, We call the resulting set of pages, “KELVINpedia”, as each page (*i.e.,* KB entry) looks much like a Wikipedia Infobox, complete with hyperlinks to other entities and to text documents that support assertions.

We have also developed a program to convert a knowledge base in TAC submission format to RDF (Lassila and Swick, 1998) using RDF reification to attach provenance and probability data to the RDF triples. This allows use of RDF-based tools to query (Prud’Hommeaux and Seaborne, 2008) and browse the data.

[9] From Washington Post article (WPB_ENG_20100506.0012 in LDC2011T07)

[10] Ian is the only computer scientist we learned about in processing an entire year of news. In contrast, the system found 52 lobbyists. We are unsure if this is a bias in our system, or if there is a larger societal message.

runid	P	R	F1	P	R	F1	P	R	F1
hltcoe1	0.5256	0.2843	0.3690	0.1620	0.2182	0.1859	0.2844	0.2551	0.2690
hltcoe2	0.4929	0.3031	0.3753	0.1818	0.2406	0.2071	0.2969	0.2755	0.2858
hltcoe3	0.4865	0.5000	0.4932	0.1849	0.3510	0.2423	0.3075	0.4342	0.3600
hltcoe4	0.4799	0.5155	**0.4971**	0.1842	0.4168	**0.2555**	0.2950	0.4718	**0.3631**
hltcoe5	0.4937	0.3053	0.3773	0.1453	0.2531	0.1846	0.2531	0.2823	0.2669

Table 7: Precision, recall, and F1 for 0-hop slots (left columns), 1-hop slots (middle columns), and 0 and 1 hops (right columns).

5 Conclusions

Our team from the JHU HLTCOE participated in both the TAC Knowledge Base Population tasks this year: Entity Linking and Cold Start. For both, we developed new systems, CALE (Context Aware Linker of Entities) for Entity Linking and and KELVIN (Knowledge Extraction, Linking, Validation, and INference) for Cold Start. Given the new status of each project, many of their components were relatively simple modules that we plan to improve.

Acknowledgments

We are grateful to BBN and CUNY for making their systems available to us.

References

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.

T. Brants. 2000. TnT. a statistical part-of-speech tagger. In *Sixth Applied NLP Conference (ANLP00)*, pages 224–231, USA.

Z. Chen, S. Tamang, A. Lee, X. Li, W.P. Lin, J. Artiles, M. Snover, M. Passantino, and Heng Ji. 2010. CUNY-BLENDER TAC-KBP2010 Entity Linking and Slot Filling System Description. In *Text Analytics Conference (TAC)*, Gaithersburg, Maryland, November.

Z. Chen, S. Tamang, A. Lee, X. Li, and H. Ji. 2011. Knowledge Base Population (KBP) Toolkit @ CUNY BLENDER LAB Manual.

Juri Ganitkevitch, Yuan Cao, Jonathan Weese, Matt Post, and Chris Callison-Burch. 2012. Joshua 4.0: Packing, PRO, and Paraphrases. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 283–291, Montréal, Canada, June. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *MT Summit*.

O. Lassila and R.R. Swick. 1998. Resource description framework (RDF) model and syntax specification. Technical report, World Wide Web Consortium.

James Mayfield and Tim Finin. 2012. Evaluating the quality of a knowledge base populated from text. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*. Association for Computational Linguistics.

P. McNamee, J. Mayfield, D.W. Oard, T. Xu, K. Wu, V. Stoyanov, and D. Doermann. 2011. Cross-Language Entity Linking in Maryland During a Hurricane. *Proceedings of Text Analysis Conference (TAC)*.

Paul McNamee. 2010. HLTCOE efforts in entity linking at TAC KBP 2010. In *Text Analysis Conference (TAC)*, Gaithersburg, Maryland, November.

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

Valentin I. Spitkovsky and Angel X. Chang. 2012. A cross-lingual dictionary for English Wikipedia concepts. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*.

V. Stoyanov and J. Eisner. 2012. Minimum-Risk Training of Approximate CRF-Based NLP Systems. In *Proceedings of NAACL-HLT*.

V. Stoyanov, A. Ropson, and J. Eisner. 2011. Empirical risk minimization of graphical model parameters given approximate inference, decoding, and model structure. *Proceedings of AISTATS*.

V. Stoyanov, J. Mayfield, T. Xu, D.W. Oard, D. Lawrie, T. Oates, T. Finin, and B. County. 2012. A context-aware approach to entity linking. *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction, NAACL-HLT*.

D. Weiss and B. Taskar. 2010. Stuctured predictions cascades. *Proceedings of AISTATS*.