Multi-lingual Extraction and Integration of Entities, Relations, Events and Sentiments into ColdStart++ KBs with the SAFT System

Hans Chalupsky[1], **Jun Araki**[2], **Eduard Hovy**[2], **Andrew Hsi**[2],
Zhengzhong Liu[2], **Xuezhe Ma**[2], **Evangelia Spiliopoulou**[2], **and Shuxin Yao**[2]

[1]USC Information Sciences Institute, Marina del Rey, CA, USA
[2]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

This paper describes our participation in the TAC-KBP 2017 Cold Start++ Knowledge Base Population task. Our SAFT system is a loosely-coupled integration of individual components processing documents in English, Spanish and Chinese. The system extracts entities, slot relations, event nuggets and arguments, performs entity linking against Freebase and event coreference, and also integrates sentiment relations extracted by external collaborators at Columbia and Cornell. The various extractions get combined, linked, deconflicted and integrated into a consistent knowledge base (one per language) for query-based evaluation.

1 Introduction

This paper describes our participation in the TAC-KBP 2017 Cold Start++ Knowledge Base Population task. Our SAFT system is a loosely-coupled integration of individual components processing documents in English, Spanish and Chinese. The system extracts entities, slot relations, event nuggets and arguments, performs entity linking against Freebase and event coreference, and also integrates sentiment relations extracted by external collaborators at Columbia and Cornell. The various extractions get combined, linked, deconflicted and integrated into a consistent knowledge base (one per language) for query-based evaluation.

The 2017 Cold Start++ KB population task was a very significant extension of Cold Start KBP (CSKBP) tasks held in previous years along a number of dimensions:

(1) It was mandated to be tri-lingual for English, Spanish and Chinese. While the 2016 CSKBP task was also organized for all three languages, participants were free to choose which language conditions to participate in.

(2) Cold Start++ fully integrated the entity discovery and linking aspect which previously was evaluated separately. A CSKBP system will generally need an entity linker to perform its task, but in the past performance was only evaluated relative to slot-filling queries. This time linking of entities to the Freebase reference KB was also evaluated.

(3) Cold Start++ fully integrated a tri-lingual version of the event nugget detection and linking task held in previous years, as well as

(4) a tri-lingual version of the event argument detection and linking tasks from 2015 and 2016.

(5) Finally, a tri-lingual sentiment detection task was added, however, this task was somewhat simplified relative to the full BeSt tasks organized in prior years and only focused on sentiment relations between person entities.

Needless to say, Cold Start++ was extremely challenging. A mono-lingual CSKBP participation is a difficult task to complete. In addition, Cold Start++ added EDL, event nuggets, event arguments and sentiments across three languages which literally increased the task complexity by a factor of 10 or more.

Our team had a strong technology base to start with particularly for EDL and events and was working towards a full participation in all dimensions of the Cold Start++ task. In the end, however, we ran out of time and fell short with respect to (1) Chinese slot filling relations where we finished and ran the extractor but failed to integrate its results, and (2) Spanish slot filling relations where various training data preparation and preprocessing had been finished, but we failed to complete and run the extractor on the Spanish document set.

2 SAFT Cold Start++ System Architecture

Figure 1 shows the overall architecture of our SAFT Cold Start++ KBP system for the TAC-KBP 2017 evaluation. The system is an asynchronous, distributed, loosely-coupled integration of modules that generally communicate by exchanging files. Inputs and outputs for most modules are segmented by document, or are otherwise concatenations of per-document outputs, as is the case for the EDL component. File formats are either native formats used by components such as CoreNLP or the Joint LSTM parser (e.g., CoNNL), standard output formats defined by TAC (for example, the formats used by EDL and the KResolver Mini-KB outputs), or minor variations of TAC-KBP formats (e.g., for the various event nugget, argument and coreference components).

Modules were run by different team members at different sites, and upon run completion result files were archived and manually shipped to other team members to allow their modules to run using those results as inputs. Each component processes its inputs fully automatically, some then send their results automatically to downstream components (e.g., from English Slot Relations to KResolver Mini-KB Integration), while others currently require manual data exchange (for example going from the Event Merger to KB Integration). Running in this distributed fashion allowed us to leverage existing installations and computing infrastructure at different sites with minimal migration and installation overhead; however, there is no principal restriction to this mode of operation, everything could have been run fully automatically end-to-end with some extra engineering overhead.

Connections between modules indicate input-output dependencies. For example, EDL requires document pre-processing by CoreNLP. Line and box colors indicate language-specific data flow and processing capabilities. For example, Event Nuggets A takes English (black) and Chinese (red) preprocessing from CoreNLP and produces English and Chinese event nuggets which are forwarded to Event Coref A. Similarly, Event Nuggets B takes English (black) and Spanish (blue) as inputs and produces English and Spanish event nuggets. All components have access to raw input text which is not shown except for Event Nuggets B which does not require any other significant preprocessing. The picture is still somewhat simplified, since multi-language modules are not always uniform in their capabilities across languages (e.g., CoreNLP provides a reduced set of models for Spanish). Moreover, CoreNLP which was used by a number of different SAFT modules was run multiple times at different sites with different configurations.

All modules shaded in gray originated from within the SAFT team. Third-party modules such as CoreNLP and the Malt parser were run by SAFT team members as needed to drive their own modules. The only exception are sentiment relation extractors which were run by external contributors not part of the SAFT team, based on inputs provided by SAFT. Dotted lines indicate incomplete modules or connections as for (1) Chinese Slot Relations where we finished the module and ran the extractor but failed to integrate its results, and (2) Spanish Slot Relations where various training data preparation and preprocessing had been finished, but we failed to complete and run the extractor on the Spanish document set.

Below we provide more detail on individual component modules and their respective performance. There are generally four kinds of evaluation results reported: (1) composite results from the full, query-level evaluation of the resulting KBs, (2) component-level results where performance of individual modules was measured from the result KBs by comparing to a comprehensive gold standard for a small subset of documents, (3) standalone results where modules were evaluated in one of the standalone tracks of the 2017 evaluation, and (4) other individual module evaluation results in case no other results are available.

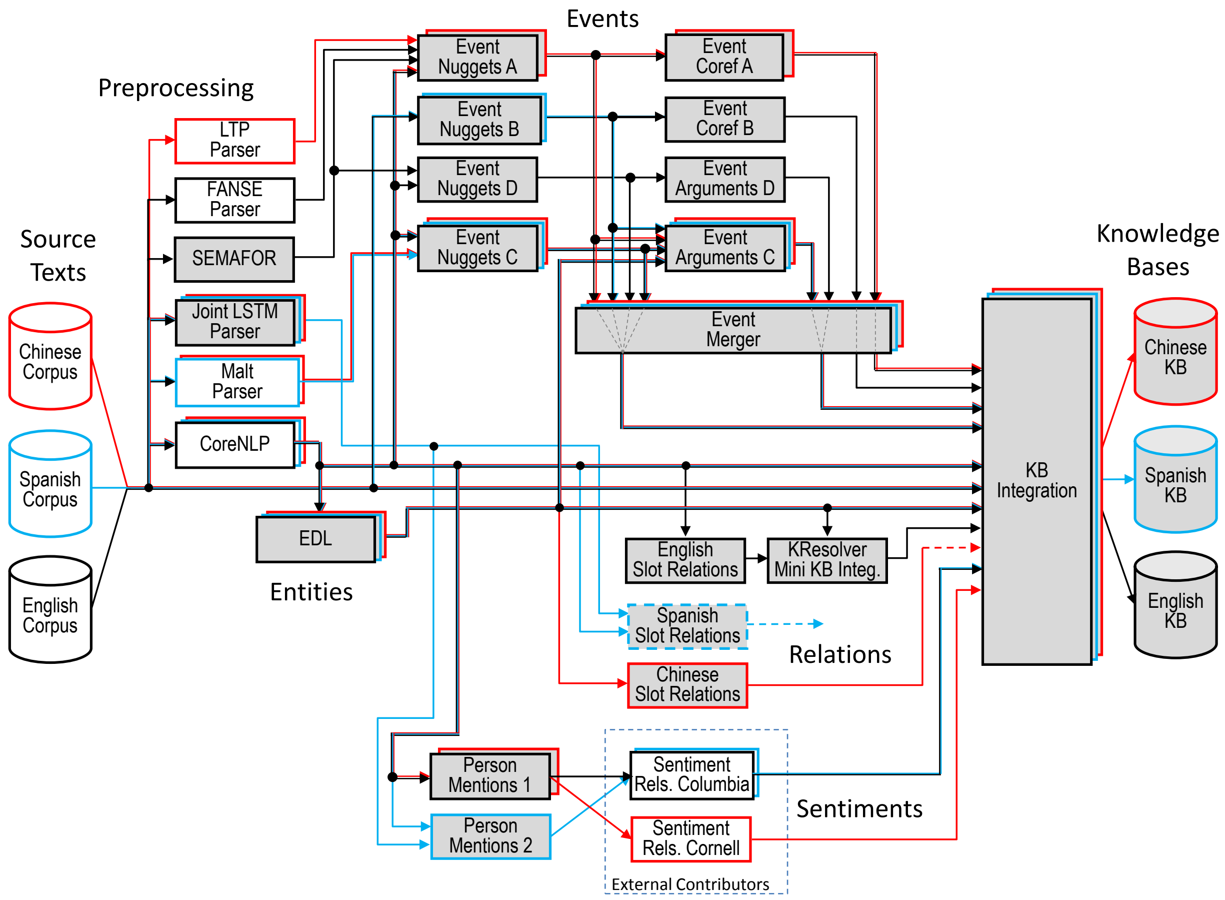

Figure 1: System architecture

3 Entity Discovery and Linking

Our tri-lingual Entity Discovery and Linking module (EDL) is based on a system we developed for the TAC-KBP 2015 EDL track, and since then extended and improved to address new challenges added for recent EDL tracks (Fauceglia et al., 2015; Fauceglia et al., 2016). In particular, the 2016 and 2017 tracks targeted larger-scale data processing by increasing the size of source collections from 500 to 90,000 documents, and expanded targeted individual nominal mentions from only person mentions for English (e.g., “the president”) to all entity types and to all three languages (e.g., “the city” or “la compañía”).

Our end-to-end EDL system includes XML document file parsing, entity extraction, linking, type inference and NIL clustering. We use the Stanford CoreNLP pipeline (Manning et al., 2014) for preprocessing and named entity recognition, and adapt and extend (Moro et al., 2014) for entity extraction and linking. Our system first processes all of Wikipedia, representing it as a directed weighted graph and then computes a semantic signature for each vertex. Second, we use these semantic signatures for entity discovery and linking across three languages in a system that uses an extended version of Babelfy[1] as its backbone. The system is described in more detail in (Ma et al., 2017), below we just briefly describe the main processing steps and evaluation results.

Our system first constructs a directed weighted *graph of Wikipedia*, where vertices represent entities and concepts in Wikipedia. An edge exists from vertex $v1$ to $v2$ if $v2$ appears in $v1$’s page as a text anchor. Following Moro et al. (2014), the weight of each edge is calculated as the number of triangles (cycles of length 3) that this edge belongs to. To implement the graph, we used the WebGraph framework (Boldi and Vigna, 2004).

[1] http://babelfy.org

	NER			Linking			Clustering		
	P	R	F1	P	R	F1	P	R	F1
Eng	79.3	52.3	63.0	69.0	45.5	54.8	68.9	45.4	54.7
Spa	78.0	45.9	57.8	70.8	41.7	52.5	67.5	39.7	50.0
Chi	72.2	43.6	54.4	62.4	37.7	47.0	66.4	40.1	50.9

Table 1: Official 2017 standalone TEDL evaluation results over all three languages for our best run for three key metrics: strong typed mention match (NER), strong typed all match (Linking), and mention CEAF (Clustering).

	NER			Linking			Clustering		
	P	R	F1	P	R	F1	P	R	F1
	Named Mention								
Eng	81.4	64.5	72.0	72.8	57.8	64.4	72.6	57.5	64.2
Spa	78.0	62.5	69.4	70.8	56.7	63.0	67.5	54.0	60.0
Chi	72.2	56.6	63.4	62.4	48.9	54.8	66.4	52.0	58.3
	Nominal Mention								
Eng	60.4	15.6	24.8	33.0	8.5	13.5	42.5	11.0	17.4
Spa	–	–	–	–	–	–	–	–	–
Chi	–	–	–	–	–	–	–	–	–

Table 2: Official standalone TEDL results on named and nominal mentions, respectively.

Using the completed Wikipedia graph, we compute a *semantic signature* for each vertex, namely the set of other vertices strongly related to it. Relation strengths are computed using Personalized PageRank with node-dependent restart (Avrachenkov et al., 2014) (which differs from Babelfy). Vertices with a score lower than a threshold are discarded.

Different from last year's system, this year we used the pre-trained NER system implemented in Stanford's CoreNLP to extract mentions, which significantly reduced the number of mentions and accelerated the linking algorithm. Given these mentions we perform *candidate extraction* by searching through Wikipedia for candidate entities for which (one of) the names of the entity is a superstring of the text of the named mention. For nominals, we focus on named mentions within a certain window size around the nominal and exploit a special word/entity embedding we trained for this task to find candidate entities that are similar to and coherent with the head word of the nominal.

After the above steps, a *semantic interpretation graph* is constructed by uniting all semantic signatures of every candidate. A graph densification algorithm is then applied iteratively until a density threshold is reached. Finally, every mention is linked to its most likely candidate (including NIL) according to a scoring function.

Once a candidate Wikipedia entity is found, it is mapped back onto Freebase via a map built beforehand and we perform *type inference* based on its type specifications in Freebase using a small set of rules. If a candidate entity has one of the five target types, it is assigned that type, otherwise it will be discarded. The final step is *NIL clustering* where we simply merge candidates with exactly the same name spelling.

Table 1 shows results from the 2017 standalone TEDL evaluation for all three languages for our best run for three key metrics: strong typed mention match (NER), strong typed all match (Linking) and mention CEAF (Clustering). Table 2 shows results broken out for named and nominal mentions, respectively. It should be noted that our system can currently only handle English nominal mentions.

4 Event Extraction Systems

While prior CSKBP tasks focused exclusively on the extraction of entities and slot relations (e.g., `per:spouse`), a major goal of the 2017 Cold Start++ task was the integration of event information. The following aspects of events are addressed by the task: (1) Event nuggets, which are trigger words or phrases indicating an event such as "killed" or "election". (2) Event arguments that fill in one or more roles of an event such as the victim of a killing or the person elected in an election. (3) Event coreference within-document which groups events and their arguments into Rich ERE event "hoppers" (Song et al., 2015), as well as cross-document event coreference which links hoppers across documents. (4) Realis indicating whether an event actually occurred or whether it is generic, unspecific, failed, future tense, etc.

All these aspects were evaluated in standalone event evaluations this year and in prior years, and their integration into Cold Start++ follows the guidelines from those standalone evaluations. The only exception is cross-document event coreference which is new to Cold Start++ imposed by treating the result as a linked knowledge base. However, the challenge of this cross-document event coreference requirement was softened somewhat by stipulating that events will never be the initial or intermediate subject of evaluation queries.

Our team has a rich portfolio of event extraction systems from prior TAC-KBP event evaluations which were further extended and refined for Cold Start++. Those systems are labeled A-to-C in Figure 1. In addition, a new system D was developed this year to provide additional nugget and argument detection for English as well as a merging component to integrate our rich set of event processing results. Below we describe these subsystems in some more detail, for additional information see (Liu et al., 2016; Hsi et al., 2017; Spiliopoulou et al., 2017).

4.1 Event Nuggets and Coreference System A

The goal of this module is to detect event nugget instances and coreference clusters that group together the nuggets referring to the same underlying events. The targeted languages are English and Chinese. The nugget detector is an extension of systems developed for prior TAC-KBP evaluations (Liu et al., 2015; Liu et al., 2016). It employs a Conditional Random Field (CRF) model (Lafferty et al., 2001) trained discriminatively using the Passive-Aggressive algorithm. We use a number of tools for preprocessing, syntactic as well as semantic parsing: CoreNLP, SEMAFOR (Das et al., 2014), FANSE Parser[2] as well as the LTP[3] parser and toolkit for Chinese.

The following traditional linguistic features are used for both languages:

- Lemma, part-of-speech (POS), named entity tags of the trigger itself, and the words in a 2-word window (both sides)
- Lemma, POS of the two bigrams that include the trigger
- Brown clusters, WordNet synonyms and derivative forms of the trigger
- Selected WordNet senses of tokens in the trigger's sentence
- Closest named entity type
- Lemma, dependency type and POS of the child and head of the trigger based on dependency
- Frame name and semantic argument features (lemma, POS, NER tag) from semantic parses

We also include character level features for Chinese, including the containing characters of the trigger, the first/last character of the trigger, and the head character configuration structure (that is, at which position did the head character appear).

Our coreference resolver is a Latent Antecedent Tree model that constructs a tree based on the detected nuggets using three types of features:

- Trigger match: exact and partial match of trigger words
- Argument match: exact and partial match of the arguments
- Discourse features: sentence and mention distances

Matching between coreference candidates is based on word vector similarity, Brown cluster matching, WordNet sense matching, POS, lemma, mention type and mention realis. Event nugget and coreference results are then merged with those from other CMU SAFT team systems to create a merged output (see Section 4.5).

Due to the merging of multiple event module outputs, we do not have individual results for this module alone. Our component-level results (detailed in Table 7) are F1=35.85 mention span detection score

[2] http://www.isi.edu/publications/licensed-sw/fanseparser/

[3] http://ltp.ai

for English, which is the top result for all Cold Start++ teams. Our coreference score is 13.56 in terms of KBP average, which ranks at second place for English. Our Chinese systems performance is slightly lower at F1=29.06, and the coreference KBP average is 8.71. Note that most individual KBP participants have a higher recall than precision, while we observe the opposite. This is likely due to the fact that we merge the results of multiple systems.

4.2 Event Nuggets and Coreference System B

Similar to System A, the goal of event nugget detection system B is to identify event spans in text and assign an event type to each detected span. The targeted languages are English and Spanish. System B can detect 38 event types as defined for the TAC KBP 2015 Event Nugget task, which is a superset of the 18 event types used in Cold Start++. We formalize event detection as a sentence-level sequence labeling problem using the BIO scheme, where B is the beginning of an event nugget, I is inside, and O is outside. This means that every token is classified into one of 77 classes (i.e., O, B-X or I-X where X is one of the 38 event types).

Our approach is an extension of a neural event detection system developed for TAC-KBP 2016 (Liu et al., 2016) that uses a bidirectional long short-term memory (BiLSTM) (Graves and Schmidhuber, 2005). BiLSTMs have been shown to successfully capture contextual information of a sentence or its subsequence, achieving superior performance in numerous sequence modeling tasks such as dependency parsing (Wang and Chang, 2016), relation extraction (Miwa and Bansal, 2016), sentiment analysis (Ruder et al., 2016), and question answering (Hermann et al., 2015). BiLSTMs are a variant of LSTMs (Hochreiter and Schmidhuber, 1997) that enhance standard LSTMs by modeling a sequence in both forward and backward directions with two separate hidden states to capture past and future information. Future information can be important in event detection, because event arguments are often effective features for event detection and some arguments such as patients and locations tend to appear after an event nugget in a sentence.

In sequence labeling problems such as event detection, independent token-level classification decisions are limited, and it is beneficial to consider the dependencies between labels in neighborhoods and jointly decode the best sequence of labels for an input sentence. Therefore, instead of decoding labels independently, we model them jointly using a conditional random field. Specifically, we put a CRF layer on top of BiLSTM layers, similarly to (Ma and Hovy, 2016; Lample et al., 2016). For training we use the TAC KBP 2015 event nugget dataset. We also employ pre-trained 50-dimensional GloVe word vectors (Pennington et al., 2014) and do not fine-tune them during training. We use Adam (Kingma and Ba, 2015) to optimize parameters based on the performance in the validation dataset. Our experiments show that the model achieves 61.90 F1 in span detection and 55.91 F1 in span+type detection on the TAC KBP 2015 test data (English). This performance is close to the state-of-the-art, and it was ranked third in the official results of TAC KBP 2015.

4.3 Event Nuggets and Arguments System C

The primary goal of this module is to extract event arguments for each event discovered by the various SAFT event nugget modules. The targeted languages are English, Spanish and Chinese. The system is described in more detail in (Hsi et al., 2017), here we only briefly summarize its main characteristics.

The overall pipeline for event argument extraction is as follows: We begin by performing preprocessing using Stanford CoreNLP and the MaltParser[4] on the input documents to extract information such as tokenization, part of speech tags and dependency parses. We then obtain entity extractions from two different sources: (1) a model trained using the standalone Stanford NER tool, and (2) the EDL output from the module described in Section 3. We then obtain event nugget information from (1) the CRF-based event nugget detection System A described in Section 4.1 designed for English and Chinese, (2) the BiLSTM-CRF-based event nugget detection System B described in Section 4.2 designed for English and Spanish, and (3) a logistic regression classifier applied to each word in the document designed for all three languages (labeled Event Nuggets C in Figure 1).

The output from entity extraction and nugget de-

[4] http://www.maltparser.org/

	Arg Score	Link	P	R	F1
Eng	2.53	1.76	21.99	6.84	10.44
Spa	1.56	0.38	31.45	1.95	3.67
Chi	4.00	1.71	28.84	7.82	12.30

Table 3: Official 2017 standalone event argument and linking results for Event Arguments System C for the following metrics: error-based argument score, B^3-based linking score (both at the median of the confidence interval), and general precision, recall and F1 for argument tuple extraction.

tection is then fed into a logistic regression argument classifier, which makes predictions of argument relationships on every entity/nugget word pair within the same sentence. Finally, a realis label is predicted for each discovered argument, once again using logistic regression. For training, we used the ACE 2005 and RichERE datasets. Word embeddings for all three languages were obtained from their respective Wikipedia dumps using word2vec. Arguments are extracted separately for each set of event nuggets coming from systems A, B and C and are then merged by the Event Merger (see Section 4.5).

Our logistic regression classifiers use a combination of language-dependent features (e.g. lexical features, embeddings, language-specific part-of-speech tags) and language-independent features (e.g. Universal POS tags, Universal Dependencies, entity type information). This enables us to train a single cross-lingual model that can be applied to all three target languages. The effect of our cross-lingual training is most noticeable when there is little annotated event training data available (as is the case for Spanish).

Table 3 summarizes the evaluation results for this component on the TAC-KBP 2017 standalone event argument extraction task. Details of our participation and results are described in (Hsi et al., 2017).

4.4 Event Nuggets and Arguments System D

The goal of this module is again event nugget detection together with the extraction of any of their arguments for each event discovered. The targeted language for this module is English only.

The main idea behind our event detection approach is that frame-semantic parsers generate a rich set of predicates that can directly serve as event nuggets. To this end, our approach starts with the output of a frame-semantic parser which is then refined in order to get a large set of event nugget candidates. This allows us to exploit the rich semantic structure generated by such a parser to generate more event candidates and achieve higher recall than previous systems. Our approach is described and evaluated in more detail in (Spiliopoulou et al., 2017), below we just briefly list its main characteristics.

In order to generate a list of candidate events, we use SEMAFOR (Das et al., 2014) which is a frame-semantic parser based on FrameNet (Fillmore et al., 2003) that links terms and text spans to frames and their roles as defined by FrameNet. Since FrameNet covers a wide range of semantic structures including events, entities, time units, and many more, filtering and refinement is necessary to focus on events only. To do this we utilize structural similarities between FrameNet and the TAC KBP ontology. We observe that most types of the TAC KBP event ontology can be decomposed into a small set of FrameNet frames. Thus, we first manually construct a many-to-one mapping from a subset of FrameNet frames to TAC-KBP event types. For example, any of the frames `Attack`, `Destroying`, `Downing`, `Explosion`, `Hostile encounter`, `Invading`, `Shoot projectiles`, or `Use firearm` might indicate a TAC-KBP event type of `Conflict.Attack`. We then detect our event nuggets based on this mapping: a mention generated by SEMAFOR is accepted as an event only if its frame is in the domain of the mapping.

The final part of the system involves the extraction of arguments for all extracted event nuggets. For this part we decided not to use the frame-semantic parser, since FrameNet's frame roles have very different definitions from the argument roles described in the TAC KBP guidelines. Instead, we use the dependency graphs produced by Stanford's CoreNLP parser in order to assign most of the arguments of an event nugget. Specifically, for location and time arguments, we used the CoreNLP NER module and we assign a named entity as time or location of an event nugget only if both occur at the same sentence.

4.5 Event Merger

This module combines the outputs of the four event detection systems described above. It generates a set of merged event nuggets for all three languages, a set of merged event arguments for all three languages and it integrates the event coreference clusters from Systems A and B into a uniform format but leaves them unmerged.

The module first takes the union of the collected outputs (nuggets and arguments) and then runs a neural net classifier to provide confidence scores for each event nugget instance. These confidence scores depend only on the type of event that each system predicts for every candidate event (which is `None` if a system did not classify a certain mention as an event). We currently do not have a mechanism to compute confidence values for event arguments.

5 Relation Extraction Systems

The relation extraction systems shown in the lower part of Figure 1 extract the 65 TAC-KBP Cold Start slot relations used in past CSKB evaluations. These slots divide into 15 string-valued slots such as `per:title` or `org:website`, 26 entity-valued slots such as `per:spouse` or `org:founded_by`, plus an additional 24 inverse slots added specifically for CSKB evaluations to make all entity-valued slots traversable in both forward and backward directions.

Our initial portfolio of Cold Start relation extraction systems was much smaller compared to our large number of event modules, and consisted only of a single limited-coverage English relation extraction system used in the 2015 English CSKB evaluation (Chalupsky, 2015). Therefore, to address this part of the task, we had to extend and improve the existing English extractor and build new extractors for Spanish and Chinese from scratch. The challenge for Spanish and Chinese which are relatively recent additions to the task is the small amount of directly relevant training data available from previous evaluations. We addressed this challenge by using a machine translation approach for Spanish and a distant supervision approach for Chinese described in more detail below.

5.1 English Slot Relations

The first goal of this module is to detect relation arguments and any of the 65 CSKB slot relations that hold between them. A second goal is to link entity-valued relation arguments to a descriptive name within the document for cases where an argument is a pronoun or nominal. These names are then used by KB integration components in conjunction with EDL results to link relations into a KB.

Our English relation extractor extends a limited-coverage extraction system we built for the 2015 CSKB evaluation (Chalupsky, 2015) called Knowledge Resolver (or KRes). The system uses (1) a pattern-based extractor for a subset of the relations, (2) an extended full-coverage statistical extractor, and a name-linker that uses a small set of dependency patterns in conjunction with coreference information from CoreNLP.

KRes is a logic-based inference system based on the PowerLoom[5] knowledge representation and reasoning system aimed at improving relation extraction through the exploitation of richer semantic information. KRes uses the Stanford CoreNLP toolkit for tokenization, POS-tagging, sentence detection, NER-typing, dependency parsing and coreference resolution. CoreNLP annotations (such as sentences, mentions, NER-types, parse trees, etc.) are then translated into a logic-based data model for the PowerLoom KR&R system.

The *pattern-based extractor* is very similar to previous versions and described in more detail in (Chalupsky, 2013; Chalupsky, 2014). It applies a set of dependency patterns represented as PowerLoom terms to the various annotations generated by CoreNLP. We developed patterns for the following nine TAC-KBP slot relations: `per:age`, `per:children`, `per:employee_or_member_of`, `per:other_family`, `per:parents`, `per:siblings`, `per:spouse` and `per:title`. Each pattern match identifies two relation arguments as well as the detected relation type between them.

The *statistical extractor* is an extension of previ-

[5] http://www.isi.edu/isd/LOOM/PowerLoom/

ous versions that (1) now addresses the full set of TAC-KBP slots, (2) uses a single multi-class classifier instead of the binary classifiers used before, (3) does not make use of features based on SEMAFOR anymore, and (4) adds some new features such as Brown clusters compared to what is described in (Chalupsky, 2014).

The extractor starts by detecting possible arguments of types relevant to TAC-KBP slots. Argument mentions and their types are constructed from NER-types detected by CoreNLP, Wordnet, as well as gazetteers such as title lists. It then enumerates possible argument pairs within a certain maximum distance in the dependency tree and then classifies each pair using a 30-class maximum entropy classifier. To keep the set of classes as small as possible we normalize each relation onto its canonical forward form and combine the various city/state/country slots onto place slots such as `place_of_residence` which are then refined later based on a more fine-grained classification of their arguments. The classifier was trained on a set of about 8,000 examples derived from previous TAC-KBP evaluations and manually inspected for errors, as well as comprehensive ERE document annotations provided by LDC.

The *name linking* component is more or less identical to previous versions and described in more detail in (Chalupsky, 2013).

The result of this extraction process is a set of typed, sentence-level relation mentions whose arguments might be named mentions, nominals, pronouns or values such as ages. Additionally, we have a set of name links connecting relation mention arguments to the best named mention describing them (where possible). We do not have evaluation results available for this module alone. Relevant composite and component evaluation results for English slot relations are described in Section 8.

5.1.1 KResolver Mini-KB Integration

The second phase of English slot relation extraction is what we call Mini-KB generation, which produces consistent per-document KBs in TAC-KBP format for each document in the corpus. These document-level mini-KBs are then combined into a global raw KB which is then further refined and deconflicted (see Section 7).

The main advantage of this scheme is scalability, since it allows us to use more expensive inferencing on a smaller, focused, per-document basis, which in addition can be performed in parallel, since documents can be processed independently. The disadvantage is that it prevents us from performing more fine-grained adjudication of conflicts when looking across documents.

The Mini-KB integration phase takes entity mentions and relation mentions generated during the relation extraction phase together with equivalence information from CoreNLP coreference, name links and EDL cross-document coreference as input. It then links equivalent entity mentions into entities and equivalent relation mentions into relations which then form an initial raw knowledge base. The challenge is that all mention-level information is noisy, incomplete, redundant, fully or partially overlapping and possibly inconsistent.

In particular, once equivalences are introduced, type information from equivalent mentions starts propagating which can commonly lead to conflicts. For example, the text "Los Angeles mayor Antonio Villaraigosa..." might generate a relation mention of type `org:top_members_employees` between "Los Angeles" and "Antonio Villaraigosa". The domain type of the relation would imply "Los Angeles" to be of type `ORG` which would conflict with a `GPE` type from EDL or a `LOCATION` type from CoreNLP for the same mention.

To address this in a principled way, we implemented an incremental KB linking, evaluation and refinement architecture. In this architecture, all annotations coming from text extraction components are treated as separate *hypotheses*. Specifically we generate *instance hypotheses* representing instances of arbitrary types implied by mention texts, *type hypotheses* for the possible types of those instances, *relation hypotheses* to represent relation mentions between instances and equivalence hypotheses to represent various instance equivalences from coreference, name links and mention overlaps.

In this phase we also map different type systems used by different extraction components onto a shared ontology rich enough to represent all necessary distinctions. For example, a `LOCATION` type from CoreNLP really means *named* location and generally corresponds to `GPE` from the TAC-KBP ontology. In this phase we also generate a more fine-grained classification of named locations into cities, states and countries. If a location mention has been linked to Freebase by EDL, we derive its narrower type from the corresponding Freebase entry. For unlinked mentions, we use a set of city, state and country gazetteers derived from the GeoNames[6] database.

In the linking phase we perform incremental *what-if* analyses of subsets of these hypotheses to see which combinations lead to conflicts and what the culprits of these conflicts are. This part of the system heavily leverages PowerLoom's multi-contextual reasoning as well as its explanation system. We currently use a greedy scheme that starts by asserting all type and relation hypotheses in a hypothetical reasoning context. We then query for all type and constraint violations, and for each violation found we analyze the proof tree to find the set of extraction hypotheses underlying the conflict. This allows us to easily exploit type and argument constraint rules (e.g., anti-reflexivity) as well as domain rules, for example, about family relations.

We then retract the weakest hypothesis in a conflict support to remedy the conflict. Hypothesis strengths are based on classifier confidences or heuristics where those are not available (e.g., relation hypotheses are generally weaker than entity type hypotheses). Next we assert mention overlap equivalence hypotheses and repeat this process, then the same for name links, coref links implied by EDL and then general coref links from CoreNLP. This process introduces noisier and noisier information at each stage and then retracts the weakest hypotheses underlying any newly discovered conflicts. At the end we use the set of surviving type, relation and equivalence hypotheses to form the mini-KB for the current document.

Finally, we translate entity and relation hypotheses from our intermediate integration ontology into the TAC-KBP type system. For example, place relations such as `hasPlaceOfDeath` together with more fine-grained types such as `City` for the second argument translate into `per:city_of_death`, too fine-grained family relation such as `hasNiece` are mapped onto `per:other_family`, and we also perform some other inferences for inverse slots and inferring employment from top employment. We additionally perform some value normalization here, e.g., for ages and dates, however, normalization for place names is still missing, which accounts for some redundancy and inexact match errors.

Next we output a mini-KB file in TAC-KBP KB format for entities, types and relations with associated provenance. We do not yet eliminate redundancies which is left to Phase 3 of the KB construction process (see Section 7). Most importantly, entities linked to Freebase m-IDs or NIL clusters by EDL receive KB IDs based on these identifiers, which will automatically link them with corresponding entities from other documents, thus, forming a globally linked knowledge base.

5.2 Spanish Slot Relations

The Spanish Slot Relations module's primary goal was to extract the 65 CSKB slot relations from Spanish documents. Given the relatively short amount of time and limited man power available to us, we aimed at building an extractor very similar in structure to the system we had previously built for English described in the previous section. To do this there were two primary challenges we needed to address: (1) a very limited amount of available training data, and (2) a more restricted set of NLP tools and resources available for Spanish.

Our approach for the first challenge was to use machine translation from Microsoft's Azure free tier machine translation service. However, instead of translating source documents from Spanish into English and then running our English extraction pipeline over it (which would have been cost-prohibitive for a corpus of 30,000 documents), we decided to translate our corpus of English relation annotations described above, which could easily be done using Microsoft's free tier service. Relation annotations need precise delineation of argument spans which get lost in plain translated output. Fortunately, the Azure service can handle HTML markup in its input and tries to preserve tags and their logical positions in the translated output. This allowed us to mark up arguments in the English input and have the translated sentences marked up with their Spanish argument counterparts.

We also used Azure to translate our English

[6] www.geonames.org

gazetteers for titles, geo-names, crimes, etc. into Spanish. Additionally, we extended our gazetteers with a bootstrap approach using embeddings from the Spanish Billion Words Corpus[7] and a small amount of manual checking and filtering. To spot-check translation quality, we back-translated small samples via Google's Spanish-to-English translation service, and the results of those checks looked generally encouraging.

To address the second challenge, we had to resort to use different, less established tools that generally required significant effort to be integrated into a production pipeline. For example, since CoreNLP only supported a limited pipeline for Spanish, we instead used a Joint LSTM semantic dependency parser for Spanish developed by other members of our team (Swayamdipta et al., 2016) not involved in this evaluation. We also had to address various other issues such as lemmatization, training up Brown clusters from scratch or to procure an entity coreference system for Spanish.

In the end, a large number of these preparation and preprocessing tasks were finished, but we ran out of time and failed to complete and run the extractor on the Spanish document set. For this reason, the module box in Figure 1 is drawn with dotted lines and we do not have any Spanish evaluation results for queries that involved any slot relations.

5.3 Chinese Slot Relations

Our Chinese relation extraction module aims to find CSKB slot relations from Chinese text. Slot relations are comprised of 41 base relations and their inverses. In order to automatically extract slot relations, we use an ensemble of rule-based classifiers and a bi-directional Gated Recurrent Unit (GRU) model with sentence-level attention.

Due to the sparsity of available training data, we used distant supervision (Mintz et al., 2009) to generate labeled training data from available knowledge bases and linkable unlabeled corpora. Specifically, we used DBpedia (Auer et al., 2007) as the KB to generate facts containing the relations of interest by manually mapping slot relations to either directly corresponding DBpedia relations or multi-hop relation paths. This resulted in 23 slot relations being mapped to at least one DBpedia relation or path. For the remaining relations we manually created rules and built a small number of rule-based classifiers to generate the final results.

DBpedia contain a very large number of real-world facts in (entity1, relation, entity2) triple format. We generated training data by aligning 208,259 relational facts extracted from DBpedia with Wikipedia articles, and assuming that if two entities participate in a relation, any sentence that contains those two entities might express that relation. In the end, we generated 1,711,341 instances for training, and 398,566 instances for testing, each instance being a sentence (possibly) expressing one of the relations mined from the KB.

There is an inevitable noisy labeling problem that accompanies distant supervision. In order to tackle that, we follow the idea of (Lin et al., 2016), (Zhou et al., 2016) and the work from the Natural Language Processing Lab at Tsinghua University[8], and use a bi-directional Gated Recurrent Unit (GRU) model with selective attention over instances for relation extraction, which can dynamically reduce the weights of noisy instances and make better use of informative ones.

We start by constructing a Chinese character embedding using a skip-gram model (Mikolov et al., 2013). All words with less than 5 occurrences are removed, numbers and dates are replaced with special tokens, and named entities are recognized and concatenated together by underscores.

For each training instance to the neural relation extraction model we have an entity pair and a set of sentences as input, and the known relation labels as output. In the first layer of the model, each character w_t in the sentence is represented as the vector

$$[v_t^{(w)}; v_t^{(p)}; v_t^{(n)}]$$

where $v_t^{(w)}$ is the character embedding, $v_t^{(p)}$ is the position embedding which encodes the relative distance from the current word to the head or tail entity, and $v_t^{(n)}$ is a vector indicating whether w_t is part of a named entity using the entity type with a BIO label.

In the second layer, a bidirectional GRU is used to encode each sentence. The hidden representations of each time step from both directions are concate-

[7] http://crscardellino.me/SBWCE/

[8] https://github.com/thunlp/TensorFlow-NRE

nated as the features of each word. We used two attention mechanisms on different granularities: word level and sentence level. Word-level attention calculates attention scores for each word in the sentence to determine which words are more important for expressing the relation. A weighted sum of the hidden representations using attention scores are used to represent the sentence. Sentence-level attention calculates attention scores for each sentence of a given entity pair to select the more informative sentences and to reduce the negative influence of label noise from distant supervision. A final softmax layer is used to calculate the probability of each relation as well as cross entropy for calculating loss.

Our approach forms a corpus-level relation extractor that predicts relations between entity pairs collectively based on all sentences in the corpus where two entities co-occur. This is different from more traditional sentence-based approaches as used, for example, by our English slot relations component. Since a Cold Start++ submission requires provenance for each extracted relation, we selected the top-3 sentences with highest attention scores for each relation prediction as its textual provenance.

Applied to the 30,000 documents from the 2017 Chinese evaluation corpus, our extractor produced 76,307 relations with confidences of at least 0.5, supported by 82,803 pieces of textual provenance (that is, most relations had only one textual support). Unfortunately, KB integration of Chinese slot relation results did not get finished in time and we therefore do not have any relevant component-level results from the evaluation. Instead we provide our internal evaluation results from applying the trained neural model to the distant supervision test set. Table 5.3 shows total Area Under the Curve (AUC) as well as precision numbers at the top-scoring 300, 600 and 900 relation instances.

Evaluation Metric	Result
AUC	0.832
Precision @300	0.97
Precision @600	0.957
Precision @900	0.944

Table 4: Internal evaluation results of Chinese slot relation extraction

6 Sentiment Extraction Systems

Our team did not build any sentiment extraction systems. Instead, we were able to enlist outside help from Columbia University for English and Spanish, and from Cornell University for Chinese, who both provided the principal sentiment extraction components for the Tinkerbell team. All sentiment extractors take document annotations in LDC's ERE format as input and produce sentiment annotations in the BeSt XML format developed during past TAC-KBP belief and sentiment evaluations.

The tri-lingual sentiment detection task of Cold Start++ was significantly simplified relative to the full BeSt tasks organized in prior years, and only focused on sentiment relations between person entities. For Cold Start++ sentiments are therfore represented only by two person-to-person `per:likes` and `per:dislikes` relations and their inverses.

Since our EDL component only produced named mentions for all three languages and nominals only for English, we decided to build specialized person mention detectors to feed the external sentiment extractors for better recall. To this end we built person mention detectors for English and Chinese based on the respective standard CoreNLP pipelines plus Wordnet for nominal mentions, and a Spanish mention detector which also used CoreNLP plus Wordnet plus our own tri-lingual JLSTM dependency parser due to the limited functionality of the CoreNLP pipeline for Spanish. For each language, specialized processing was used to include the authors from discussion forum posts. All three detectors packaged the mentions they found into per-document ERE XML files which were shipped to our external collaborators for processing. We then received corresponding per-document sentiment annotations for a subset of those mentions in BeSt format which we translated for integration into the overall KB.

7 Knowledge Base Integration

The last box in Figure 1 is the KB Integration component. It takes in all outputs from any of the extraction components across all three languages in addition to source texts and CoreNLP annotations to produce one KB file per language. In our system, KB Integration needed to address the following challenges:

(1) Cross-component linking: only KRes mini-KBs and Chinese slot relations had an initial link structure based on EDL identifiers and within-document entity coreference. Event and sentiment arguments were purely mention-based and had to be linked to global EDL identifiers where possible or otherwise unified with document-local entities from other components.

(2) KB deconflicting: as described in Section 5.1.1, once extractions are combined across components and across documents, conflicts may arise which lead to an inconsistent knowledge base. These conflicts have to be detected and resolved before the KB can be submitted for validation and scoring.

(3) KB aggregation and refining: redundant results should boost overall confidence, conflicting results should lower confidence and be resolved, duplicates should be removed and best-supported results should be reported for single-valued slots.

(4) KB and provenance formatting: the 2017 Cold Start++ KB format was extremely complex, effectively combining results from five separate TAC-KBP evaluation tasks into a single file format. Complex provenance rules, multiple justifications for the new Mean Average Precision scoring scheme, string nodes for text-valued slots, and a very large set of event-type/role combinations additionally overloaded with realis annotations made KB formatting a very significant challenge and quite a different task from previous Cold Start KB evaluations (the specification of which only became available about one month before the start of the evaluation).

An additional complicating aspect of KB integration is that it is an inherently global task that needs to take into account all or large portions of the entire corpus data at once. For this reason, it is not as trivially parallelizable as the various document-centric processing performed by individual extraction components. In the end we had between 9-11 data and result files per document and language, summing to a total of about 1 million data files that needed to be processed to build the final KBs. When it became apparent that a previously built integration component for English would not easily generalize to the new complexity and scale of the tri-lingual Cold Start++ data, we embarked on building a Python-based integration component from scratch geared very specifically to this evaluation. Unfortunately, this realization came very late in the game, and we produced about 1,500 lines of new Python code in the final two days before the submission deadline. This left only very little room for testing and led to some unfortunate bugs and surprises which are described in more detail in Section 8.

Our basic approach to KB integration was as follows: we start by building a raw KB that unions and links outputs from individual components by introducing, normalizing and merging KB node IDs as necessary. We start with EDL output which is taken more or less literally and only reformatted to conform to the Cold Start++ output format. We do not have meaningful confidences for EDL mentions, so all mentions added to the KB get confidence 1.0.

Next we output KRes mini-KB tuples (for English only) literally with the exception of relation provenance for relations that take a string value (e.g., `per:title`). These now require the introduction of a string node for the filler string plus exact provenance for the location of the filler which required some additional analysis and matching, since that provenance was not recorded as such in the mini-KB format. Since mini-KBs already have entity IDs which can be directly mapped to EDL entity IDs, nothing special has to be done for linking. Our Spanish slot relation extractor was not finished, and our Chinese extractor was finished but we did not finish the required mapping in time, so no slot relations were output for either Spanish or Chinese.

Next we output event argument results. Event mentions are mapped onto KB IDs based on their within-document coreference information which also connects to event nugget IDs. No cross-document event coreference was attempted. Cases where event mentions wound up in multiple event hoppers (e.g., due to the multiple event coreference systems we were using) were addressed by merging those hoppers. The type of an event was always based on the merged type determined by the Event Merger component. Next we tried to link event argument mentions with already existing mentions from EDL or KRes using a simple overlap match. More sophisticated methods that would also take other syntactic information and coreference into account could not be developed in time. If no linkable mention could be found, a new string entity was introduced to represent the argument. No meaningful argument confidence was produced by the event argument detector, instead we used the merged confidence for the existence of an event for this type which generally should be an overestimate for the argument confidence.

Next we output event nuggets which is fairly straight-forward. All that needs to be done here is to link them to other nuggets or events from event arguments via their within-document event coreference links. We again use merged type, realis and confidence provided by the Event Merger.

Finally, we output sentiment relations. Similar to the event arguments case, we try to match sentiment relation source and target mentions to entities introduced by EDL or KRes. If no match could be found, we introduce new document-local entities for source and/or target. We use the confidences provided by the respective sentiment extractor without any thresholding, which also included a large number of very low confidence sentiment relations.

At this point we have all the necessary information to output an initial raw knowledge base. We build this raw KB by combining all available linked information with associated provenance, and then perform some initial per-document canonicalization which maps all inverse relations onto their corresponding forward relations, then eliminates all (now redundant) inverse slots, then removes document-level duplicates and finally adds canonical mentions for each entity in a document. At this point, however, this raw KB contains a significant number of explicit and implicit conflicts. For example, we might have multiple conflicting explicit type assertions from different documents, or we might have implicit conflicts between explicit entity type assertions and implicit types implied by domain and range constraints of the various slot, event and sentiment predicates.

To remedy type conflicts, we implemented a simple majority vote system to compute a preferred type for an entity with multiple conflicting types. This system simply counts the number of explicit and implicit type judgments for each entity in the raw KB. Explicit types come from EDL and/or the KRes mini-KBs and are counted once per document. Implicit judgments come from unique domain or range types of slot, event and sentiment relations an entity participates in, and are counted once per mention. We then find problem entities with more than one type and pick a preferred type based on the counts computed before. All KB type assertions and relations that conflict with this preferred type are then simply rejected to make the KB consistent. Our majority vote system did not take confidences of types and relations into account which therefore made it vulnerable to over-valuing low-confidence information. In prior prototypes we had used strict per-component thresholding which shielded the deconflicting component from this problem. For Cold Start++, however, we retained low-confidence results to try to boost recall which lead to some unexpected and undiscovered problems discussed in the evaluation section below.

Finally, additional KB-level refinements should be performed to remove duplicates, pick best representative for single-valued relations and add inverse slots. Due to time-constraints, we did not do any further refinements along those lines and relied on NIST's Cold Start++ validator to perform them for us. KB validation attempts revealed additional issues mostly due to mentions for the same entity coming from different components with some associated provenance offset problems. These were addressed with some very specialized post-processing of the different KBs.

The resulting KBs were quite large with 1GB (8.6M lines) for English, 700MB (6.7M lines) for Spanish and 300MB (3M lines) for Chinese.

8 Evaluation Results

We submitted results from one single run per language only, each of which extracted information exclusively from the 90,000 TAC KBP 2017 Cold Start++ source documents. No other external resources were used with the exception of using Freebase to classify places into cities, states and countries after a location mention had been linked to Freebase via our EDL component.

Given the complexity of the evaluation, naturally results are also complex. This year's Cold Start++ evaluation was designed to also allow for component-based evaluations for EDL, event argument, nugget and sentiment components, based on

	NER			Linking			Clustering		
	P	R	F1	P	R	F1	P	R	F1
Eng	31.7 (-47.6)	39.9 (-12.4)	35.3 (-27.7)	19.5 (-49.5)	24.6 (-20.9)	21.8 (-33.0)	21.6 (47.3)	27.2 (-18.2)	24.1 (-30.6)
Spa	22.7 (-55.3)	28.0 (-17.9)	25.1 (-32.7)	19.9 (-50.9)	24.6 (-17.1)	22.0 (-30.5)	18.3 (-49.2)	22.5 (-17.2)	20.2 (-29.8)
Chi	38.8 (-33.4)	26.9 (-16.7)	31.8 (-22.6)	31.5 (-30.9)	21.9 (-15.8)	25.8 (-21.2)	35.0 (-31.4)	24.3 (-15.8)	28.7 (-22.2)

Table 5: The official 2017 Cold Start++ KB EDL component results of the SAFT team for all three languages for three key metrics: strong typed mention match (NER), strong typed all match (Linking), and mention CEAF (Clustering). Numbers in parenthesis show changes in comparison to TEDL evaluation results reported in Table 1.

	Arg Score	Link	P	R	F1
Eng	0.65 (-1.88)	1.07 (-0.69)	11.22 (-10.77)	5.40 (-1.44)	7.30 (-3.14)
Spa	1.35 (-0.21)	0.32 (-0.06)	31.31 (-0.14)	1.63 (-0.32)	3.10 (-0.57)
Chi	3.71 (-0.29)	1.40 (-0.31)	28.95 (+0.11)	6.89 (-0.93)	11.13 (-1.17)

Table 6: The official 2017 Cold Start++ KB event argument component results of the SAFT team for all three languages for the following metrics: error-based argument score, B^3-based linking score (both at the median of the confidence interval), and general precision, recall and F1 for argument tuple extraction. Numbers in parenthesis show changes in comparison to EAL standalone evaluation results reported in Table 3.

a subset of 500 core documents for which a comprehensive gold standard had been created by LDC. This gold standard contains 167 English, 166 Spanish and 167 Chinese documents. We start by describing component level results for each component, relate them to standalone results where available to show how KB integration affected results, and then describe our overall composite results from the query-based Cold Start++ evaluation.

Table 5 summarizes official EDL component results based on our KB submissions for all three languages. Numbers in parentheses show changes relative to the standalone results by the same component on the same documents summarized in Table 1 and described in more detail in (Ma et al., 2017). As the table shows, our EDL component results are 25-75% lower than the respective standalone results which was very unfortunate and also surprising to us. Since EDL provides the core entity structure for the KB and the entry points for the composite, query-based evaluation, having subpar entity linking significantly depresses all other results that rely on those entities. After some investigation we found the reason for this to be a deficiency in our type conflict resolution approach already hinted at above. The majority voting system did not take confidences of the underlying relations into account, which led to a large number of EDL entities having their types changed to something incorrect. For example, we had significant numbers of low probability slot and sentiment relations that led to entity type changes from GPE to ORG or ORG to PER. In previous versions of this component, confidence thresholding was done before type resolution which prevented us from discovering this problem in time.

Tables 6 and 7 summarize our official event argument and nugget component results based on our KB submissions for all three languages. For event arguments, numbers in parentheses show changes relative to the standalone results by the same component on the same documents summarized in Table 3 and described in more detail in (Hsi et al., 2017). For event nuggets we do not have official standalone results, but an internal evaluation of one of the nugget subcomponents revealed 5-10 F1-point improvements on the gold-standard documents in a standalone setting.

In general, event results are less affected by the type-deconfliction problem we described above. Event nuggets are not related to entities at all and only evaluated at the mention level as well as for their coreference to other event mentions. Event argument relations always start with an event object which does not have conflicting types from other components that it could be confused with. Only when a correct event argument gets identified with an EDL entity whose type was changed to something incorrect do we lose a correct event result due to our deconfliction problem. This is a much more uncommon situation which explains why the effects on

	Plain	Type	Realis	Type+Realis	CoNLL
Eng	35.85	28.48	25.14	20.47	13.56
Spa	16.92	11.66	12.43	9.54	5.32
Chi	29.06	23.68	22.98	19.08	8.71

Table 7: The official 2017 Cold Start++ KB event nugget component results of the SAFT team for all three languages for the following metrics: plain event mention, event type, realis, type plus realis and CoNLL linking score (all micro-averaged F1).

English	Hop 0			Hop 1			All		
	P	R	F1	P	R	F1	P	R	F1
All slots	18.69	7.93	11.13	0.24	2.52	0.44	2.61	6.76	3.77
Event only	33.62	10.46	15.95	-	-	-	-	-	-
Slot-fill only	13.36	5.92	8.20	0.24	4.43	0.46	1.22	5.57	2.00
Sentiment only	13.70	8.93	10.81	0.00	0.00	0.00	13.70	7.04	9.30
Spanish	Hop 0			Hop 1			All		
	P	R	F1	P	R	F1	P	R	F1
All slots	5.80	2.74	3.72	0.00	0.00	0.00	5.80	1.75	2.69
Event only	22.73	7.58	11.36	-	-	-	-	-	-
Slot-fill only	-	-	-	-	-	-	-	-	-
Sentiment only	3.06	3.95	3.45	0.00	0.00	0.00	3.06	3.30	3.17
Chinese	Hop 0			Hop 1			All		
	P	R	F1	P	R	F1	P	R	F1
All slots	18.78	4.17	6.82	1.94	7.08	3.05	3.89	5.09	4.41
Event only	0.00	0.00	0.00	-	-	-	-	-	-
Slot-fill only	-	-	-	-	-	-	-	-	-
Sentiment only	23.50	18.53	20.72	1.94	32.08	3.66	3.98	22.78	6.78

Table 8: The official 2017 Cold Start++ KB query-based composite results of the SAFT team for all three languages breaking out Hop-0 and Hop-1 queries as well as queries using specific slot types only (all micro-averaged LDC-MAX-ALL).

the overall argument detection F1 is less dramatic. The decrease in event nugget performance might be mostly due to the multi-component integration in the Event Merger and the conservation of lower probability results given the new MAP evaluation scheme used in this year's evaluation.

Finally, Table 8 summarizes our official query-based composite results for all three languages. The results shown use K=3 (that is the top-three results were considered if multiple justifications were given), and the LDC-MAX scoring condition which for each query picks the results from the best entry point instead of averaging over all of them. All scores are based on the Mean Average Precision (MAP) scheme used this year which took result confidences into account, and are therefore not directly comparable to results from previous years. This also means that results are generally lower for that reason alone, in addition to the complexity and additional noise coming from the multi-component integration.

Only our English KB had results for all required aspects, for Spanish and Chinese we did not finish in time with our slot relation extractors. Due to our entity type deconfliction problem, Hop-1 results are generally very low, since they always require a correct intermediate entity. For this reason, we primarily focus on Hop-0 results here. As described earlier, our event components were the most mature which is apparent in both the English and Spanish submissions. At the time of this writing, we are still investigating why event-only queries all failed for Chinese despite the fact that our event argument results for Chinese are actually the best for all three languages. For English, precision significantly dominates recall for all slot dimensions. For Spanish, results are generally very low and reflect the lack of resources and training data. For Chinese, somewhat unexpectedly, sentiment results are quite good and the best of all component-level results for all three languages.

9 Conclusion

In this paper we presented the SAFT Cold Start KBP end-to-end system used in our participation in the TAC-KBP 2017 Cold Start++ Knowledge Base Population task. Our system performed well for event nuggets, respectably for event arguments and entity discovery and linking, and relatively poorly for slot relations and overall composite query-based evaluation of the resulting KBs. There are no deep technical insights or take-aways, except that NLP pipelines are generally very complex, and that multi-linguality and the focus on multiple target modalities exponentiates this complexity. Our hope was that combining entity, event and relation extractions would provide redundancies that would improve overall results at least in some areas. This hope was squashed by the immaturity of our integration components which had to be rewritten and adjusted very late in the game which led to bugs that significantly depressed our evaluation results. If there is one key insight from all our work on this task, it is that robust integration of multi-component NLP extractions for KB generation is itself a formidable challenge that requires significant research beyond traditional NLP research vectors.

Acknowledgment

This report is based on research sponsored by the Defense Advanced Research Projects Agency (DARPA), the Air Force Research Laboratory (AFRL) under agreement number FA8750-12-2-0342. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, AFRL or the U.S. Government.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *The Semantic Web*, pages 722–735. Springer.

Konstantin Avrachenkov, Remco Van Der Hofstad, and Marina Sokol. 2014. Personalized PageRank with node-dependent restart. In *Algorithms and Models for the Web Graph*, pages 23–33. Springer.

Paolo Boldi and Sebastiano Vigna. 2004. The WebGraph framework I: compression techniques. In *Proceedings of the 13th international conference on World Wide Web*, pages 595–602. ACM.

H. Chalupsky. 2013. English slot filling with the Knowledge Resolver system. In *Proceedings of the 2013 Text Analysis Conference (TAC 2013)*. NIST.

H. Chalupsky. 2014. English slot filling with the Knowledge Resolver system. In *Proceedings of the 2014 Text Analysis Conference (TAC 2014)*. NIST.

H. Chalupsky. 2015. Cold start knowledge base population with the Knowledge Resolver system for TAC-KBP 2015. In *Proceedings of the 2015 Text Analysis Conference (TAC 2015)*. NIST.

D. Das, D. Chen, A.F.T. Martins, N. Schneider, and N.A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56, March.

Nicolas R. Fauceglia, Yiu-Chang Lin, Xuezhe Ma, and Eduard Hovy. 2015. CMU system for entity discovery and linking at TAC-KBP 2015. In *Proceedings of Text Analysis Conference (TAC 2015)*.

Nicolas R. Fauceglia, Yiu-Chang Lin, Xuezhe Ma, and Eduard Hovy. 2016. CMU system for entity discovery and linking at TAC-KBP 2016. In *Proceedings of Text Analysis Conference (TAC 2016)*.

C.J. Fillmore, C.R. Johnson, and M.R.L. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*, 16(3):235–250.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural with*, 18(5–6):602–610.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of NIPS*, pages 1693–1701.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Andrew Hsi, Jaime Carbonell, and Yiming Yang. 2017. CMU_CS_Event TAC-KBP2017 event argument extraction system. In *Proceedings of Text Analysis Conference (TAC 2017)*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL*, pages 260–270.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 2124–2133.

Zhengzhong Liu, Jun Araki, Dheeru Dua, Teruko Mitamura, and Eduard Hovy. 2015. CMU-LTI at KBP 2015 event track. In *Proceedings of Text Analysis Conference (TAC 2015)*.

Zhengzhong Liu, Jun Araki, Teruko Mitamura, and Eduard Hovy. 2016. CMU-LTI at KBP 2016 event nugget track. In *Proceedings of Text Analysis Conference (TAC 2016)*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of ACL*, pages 1064–1074.

Xuezhe Ma, Nicolas R. Fauceglia, Yiu-Chang Lin, and Eduard Hovy. 2017. CMU system for entity discovery and linking at TAC-KBP 2017. In *Proceedings of Text Analysis Conference (TAC 2017)*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*, pages 1003–1011. ACL.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of ACL*, pages 1105–1116.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2:231–244.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.

Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2016. A hierarchical model of reviews for aspect-based sentiment analysis. In *Proceedings of EMNLP*, pages 999–1005.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From Light to Rich ERE: Annotation of entities, relations, and events. In *Proceedings of the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation at NAACL-HLT*, pages 89–98.

Evangelia Spiliopoulou, Eduard Hovy, and Teruko Mitamura. 2017. Event detection using frame-semantic parser. In *Proceedings of the Events and Stories in the News Workshop*, pages 15–20. ACL.

Swabha Swayamdipta, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Greedy, joint syntactic-semantic parsing with stack LSTMs. In *Proc. of CoNLL*.

Wenhui Wang and Baobao Chang. 2016. Graph-based dependency parsing with bidirectional LSTM. In *Proceedings of ACL*, pages 2306–2315.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 207–212.