AFRL-RI-RS-TR-2018-169

SEMANTIC ANALYSIS AND FILTERING OF TEXT

CARNEGIE MELLON UNIVERSITY

JUNE 2018

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2018-169 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
ALEKSEY V. PANASYUK
Work Unit Manager

/ S /
JON S. JONES
Technical Advisor, Information
Intelligence Systems and Analysis Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
JUNE 2018	FINAL TECHNICAL REPORT	SEP 2012 – MAR 2018

4. TITLE AND SUBTITLE

SEMANTIC ANALYSIS AND FILTERING OF TEXT

5a. CONTRACT NUMBER
FA8750-12-2-0342

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62303E

6. AUTHOR(S)

Eduard Hovy

5d. PROJECT NUMBER
DEFT

5e. TASK NUMBER
12

5f. WORK UNIT NUMBER
02

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Carnegie Mellon University
Language Technologies Institute
5000 Forbes Avenue
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIEA
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2018-169

12. DISTRIBUTION AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The report describes the work of the Semantic Analysis and Filtering of Text (SAFT) project at CMU and USC/ISI funded by DARPA's Deep Exploration and Filtering of Text (DEFT). The project investigated challenging questions at the heart of multilingual semantics-oriented natural language understanding of news and similar genres. The project included six major teams/groups, focusing on: (i) semantic frame analysis, (ii) entity detection and linking, (iii) event relation extraction, (iv) event mention detection/coreference, (v) inference/relation discovery, and (vi) bringing everything together in a single multilingual knowledge base supporting English, Chinese, and Spanish. The algorithms were assembled and evaluated every year in the Text Analysis Conference (TAC) KBP and Cold Start++ KBP evaluations organized by National Institute of Standards & Technology (NIST).

15. SUBJECT TERMS

Information Retrieval, Reinforcement Learning, Dynamic Search, Domain-Specific Search

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			ALEKSEY PANASYUK
U	U	U	SAR	50	19b. TELEPHONE NUMBER (*Include area code*)

TABLE OF CONTENTS

1 SUMMARY 1

2 INTRODUCTION 1

3 METHODS, ASSUMPTIONS, AND PROCEDURES 2

3.1 Core Semantic Analysis 2
3.1.1 Final Status of This Subproject 2
3.1.2 Some Insights 6
3.1.3 Data, Code, and Other Products 7

3.2 Entity Detection and Linking 7
3.2.1 Final Status of This Subproject 7
3.2.2 Advances Made in This Subproject 10

3.3 Events: Mention (Nugget) Detection, Coreference, Script Structure 10
3.3.1 Final Status of This Subproject 10
3.3.2 Advances Made in This Subproject 14

3.4 Event Relation Extraction 14
3.4.1 Final Status of This Subproject 14
3.4.2 Advances Made in This Subproject 16

3.5 Inference/Relation Discovery 16
3.5.1 Final Status of This Subproject 17
3.5.2 Advances Made in This Subproject 17

3.6 System Integration 18
3.6.1 Final Status of This Subproject 18
3.6.2 SAFT Cold Start++ System Architecture 19

3.7 Evaluation Organization: TAC Event Mention Detection and Coreference Track 21

3.8 The EVENTS Workshop Series 21

4 RESULTS AND DISCUSSIONS 22

4.1 Modules' Performance in the 2017 KBP Task 22
4.1.1 Entity Discovery and Linking Results 22
4.1.2 Event related Results 23
4.1.3 Slot Filling Results 24
4.1.4 Sentiment related Results 30

4.2 Knowledge Base Integration 31

4.3 Full System 2017 Evaluation Results **34**

5 CONCLUSIONS **37**

6 RECOMMENDATIONS **37**

7 REFERENCES **37**

APPENDIX: PROJECT PUBLICATIONS **40**

SYMBOLS, ABBREVIATIONS, AND ACRONYMS **44**

LIST OF FIGURES

Figure 1: Example of Event and Event Nugget Output..... 12
Figure 2: Example of Event Hopper Annotations..... 13
Figure 3: Final SAFT architecture..... 20

LIST OF TABLES

Table 1 Standalone TEDL evaluation on NER, Linking, and Clustering..... 22
Table 2. Standalone TEDL evaluation on Named and Nominal Mentions..... 23
Table 3. Standalone event argument and linking results for Event Arguments..... 23
Table 4. Results of Chinese slot relation extraction (internal evaluation)..... 30
Table 5. The official 2017 Cold Start++ KB EDL component results..... 34
Table 6. The official 2017 Cold Start++ KB event argument component results..... 35
Table 7. The official 2017 Cold Start++ KB event nugget component..... 36
Table 8. The official 2017 Cold Start++ KB query-based composite results..... 36

1 Summary

The report describes the work of the Semantic Analysis and Filtering of Text (SAFT) project at CMU and USC/ISI funded by DARPA's Deep Exploration and Filtering of Text (DEFT) from 2012 to 2018. The project investigated challenging questions at the heart of multilingual semantics-oriented natural language understanding of news and similar genres. The project included six major teams/groups, each focusing on a different aspect of semantic NLP. Our researchers focused on: (i) semantic frame analysis, (ii) entity detection and linking, (iii) event relation extraction, (iv) event mention detection/coreference, (v) inference/relation discovery, and (vi) bringing everything together in a single multilingual knowledge base supporting English, Chinese, and Spanish as part of the Cold Start task. The algorithms were assembled and evaluated every year in the Text Analysis Conference (TAC) KBP and Cold Start++ KBP evaluations organized by National Institute of Standards & Technology (NIST). One track of the TAC evaluations was organized by project members (Section 3.7).

2 Introduction

Department of Defense (DoD) operators and analysts collect and process copious amounts of data from a wide range of sources to create and assess plans and execute missions. However, depending on context, much of the information that could support DoD missions may be implicit rather than explicitly expressed. Having the capability to automatically extract operationally relevant information that is only referenced indirectly would greatly assist analysts in efficiently processing data. DARPA has created the Deep Exploration and Filtering of Text (DEFT) in an attempt to move from limited, linear processing of huge sets of textual data to a nuanced, strategic exploration of available information represented by a knowledge base.

The goal of this effort was to develop a frame-based text-level semantics that spans sentences to represent the meaning of one or more related texts. Whether the input is well-structured or degraded, the output will be a graph of frames representing events and entities, related in numerous ways augmented with distributional information such as confidence is support of summarization and report writing.

In order to combine information about statistical word distributions with linguistic knowledge and represent results in semantics-oriented graph structures we focused on the following problems:

- Core semantic analysis: produce interconnected semantic graphs of richly-articulated semantic frames for sentences. Languages: English
- Entity detection and linking: identify entities in the text and link them to a given database/collection of entities. Languages: English, Chinese, Spanish.

- Event relation extraction: identify the principal event(s) in a document and extract its participants. Languages: English, Chinese, Spanish.
- Event mention detection, coreference, script structure, etc.: relate fully and partially identical events and entities. Languages: English and Chinese.
- Inference/relation discovery: discover far more relations than simple case relations and connect events and entities within the graphs with them. Languages: English.
- Cold Start KBP: integrate the extracted entity, event, and relation information into a single multilingual knowledge base. Languages: English, Chinese, Spanish.

The project contained six groups, each with faculty researchers (supported part-time, none more than 30%) and one or more graduate students:

- Semantic sentence-level analysis: Prof. Noah Smith and students Sam Thomson and Swabha Swayamdipta (with occasional funding for others).
- Entity detection and linking, using background knowledge and inference: Prof. Eduard Hovy and students Xuezhe Ma, with partial support for Nicolas Fauceglia, Yu-Chiang Lin , Qizhe Xie, Sujay Jauhar Kumar, Evangelia Spiliopoulou, and Shuxin Yao.
- Event extraction: Prof. Jaime Carbonell, Prof. Yiming Yang, and student Andrew Hsi.
- Event coreference: Prof. Teruko Mitamura and Prof. Eduard Hovy, with students Hector (Zhengzhong) Liu and Jun Araki (partial support).
- Relation discovery and event linkage: Prof. William Cohen, with student William Yang Wang and programmer Kathryn Mazaitis (partial support).
- Cold Start KBP: Dr. Hans Chalupsky

3 Methods, Assumptions, and Procedures

In this section, we describe each individual problem area and approach in more detail.

3.1 Core Semantic Analysis

Goal: Produce interconnected semantic graphs of richly-articulated semantic frames for sentences.
Languages: English
Subproject lead: Noah Smith, CMU
Principal participants: Ph.D. students Sam Thomson, Swabha Swayamdipta, Dallas Card; research staff Michael Mordowanec, Emily Danchik, Nora Kazour, Spencer Onuffer

3.1.1 Final Status of This Subproject

Our efforts in semantic analysis produced new algorithms that transform text into a wide range of output representations: frame semantics following the Berkeley FrameNet project, multiword expressions, supersenses, several variants of

dependency representations for predicate-argument semantics, and the abstract meaning representation. We also developed new approaches to constructing word embeddings (distributional lexical semantics). Here we briefly review the advances made on each front.

Frame semantics. Frame semantics is a linguistic theory that has been instantiated for English in the FrameNet lexicon. We solved the problem of frame-semantic parsing using a two-stage statistical model that takes lexical targets (i.e., content words and phrases) in their sentential contexts and predicts frame-semantic structures. Given a target in context, the first stage disambiguated it to a semantic frame. This model used latent variables and semi-supervised learning to improve frame disambiguation for targets unseen at training time. The second stage found the target's locally expressed semantic arguments. At inference time, a fast exact dual decomposition algorithm collectively predicted all the arguments of a frame at once in order to respect declaratively stated linguistic constraints, resulting in qualitatively better structures than naive local predictors. Both components are feature-based and discriminatively trained on a small set of annotated frame-semantic parses. On the SemEval 2007 benchmark data set, the approach, along with a heuristic identifier of frame-evoking targets, outperformed the prior state of the art by significant margins. Additionally, we present experiments on the much larger FrameNet 1.5 dataset. The implemented system is known as SEMAFOR. Most of this work predated the DEFT effort; the journal article compiling all of these findings was published at the beginning of DEFT (Das et al., CL 2014).

Early in the DEFT effort, we improved the implementation of SEMAFOR, including a 10x speedup, a 2/3 reduction in memory, and a significant reduction in model size, all with tiny sacrifices in accuracy.
We next focused on the task of identifying and labeling the semantic arguments of a predicate that evokes a FrameNet frame. This task is challenging because there are only a few thousand fully annotated sentences for supervised training. The approach described by Kshirsagar et al. (ACL 2015) augments an existing model with features derived from FrameNet and PropBank and with partially annotated exemplars from FrameNet. We observed a 4% absolute increase in F1 versus the original model from Das et al.

Multiword expressions. Multiword expressions (MWEs) are quite frequent in languages such as English, but their diversity, the scarcity of individual MWE types, and contextual ambiguity have presented obstacles to corpus-based studies and NLP systems addressing them as a class. In Schneider et al. (LREC 2014), we advocate for a comprehensive annotation approach: proceeding sentence by sentence, our annotators manually group tokens into MWEs according to guidelines that cover a broad range of multiword phenomena. Under this scheme, we fully annotated an English web corpus for multiword expressions, including those containing gaps.

In Schneider et al. (TACL 2014), we present a novel representation, evaluation measure, and supervised models for the task of identifying the multiword expressions (MWEs) in a sentence, resulting in a lexical semantic segmentation. Our approach generalized a standard chunking representation to encode MWEs containing gaps, thereby enabling efficient sequence tagging algorithms for feature-rich discriminative models. Experiments the aforementioned corpus of web text offer the first linguistically driven evaluation of MWE identification with truly heterogeneous expression types. Our statistical sequence model greatly outperformed a lookup-based segmentation procedure, achieving nearly 60% F1 for MWE identification.

Schneider and Smith (NAACL 2015) introduced a task of identifying and semantically classifying lexical expressions (including MWEs) in running text. We investigated the online reviews genre, adding semantic supersense annotations to the aforementioned 55,000-word English web corpus. The noun and verb supersenses applied to full lexical expressions, whether single- or multiword. We then presented a sequence tagging model that jointly infers lexical expressions and their supersenses. Results showed that even with our relatively small training corpus in a noisy domain, the joint task can be performed to attain 70% class labeling F1.

Following the efforts above, which comprised a significant portion of Nathan Schneider's dissertation, Schneider (then having moved to a position at the University of Edinburgh) co-organized the Detecting Minimal Semantic Units and their Meanings (DiMSUM) shared task at SemEval 2016 (task 10). Independently, we entered a system into this shared task (Hosseini et al., SemEval 2016). Our approach uses a discriminative first-order sequence model similar to Schneider and Smith (NAACL 2015). The chief novelty in our approach was a factorization of the labels into multiword expression and supersense labels, and restricting first-order dependencies within these two parts. Our submitted models achieved first place in the closed competition (CRF) and second place in the open competition (2-CRF).

Semantic dependencies. The task of broad coverage semantic dependency parsing aims to provide a shallow semantic analysis of text not limited to a specific domain. As distinct from deeper semantic analysis (e.g., parsing to a full lambda-calculus logical form), shallow semantic parsing captures relationships between pairs of words or concepts in a sentence, and has wide application for information extraction, knowledge base population, and question answering (among others). We participated in the SemEval 2014 Shared Task 8 on Broad-Coverage Semantic Dependency Parsing, which included datasets in three different semantic dependency formalisms, and achieved second place (Thomson et al., SemEval 2014). Our method was an arc-factored statistical model trained using conventional discriminative methods.

Peng et al. (ACL 2017) presented a deep neural architecture that parses sentences into all three of the SemEval 2014 semantic dependency graph formalisms. By using

efficient, nearly arc-factored inference and a bidirectional-LSTM composed with a multi-layer perceptron, our base system was able to significantly improve the state of the art for semantic dependency parsing, without using hand-engineered features or syntax. We then explored two multitask learning approaches—one that shared parameters across formalisms, and one that used higher-order structures to predict the graphs jointly. We found that both approaches improve performance across formalisms on average, achieving a new state of the art.

Another variant of semantic dependency parse was derived from the PropBank annotation scheme and formed the shared task at CoNLL 2008 and 2009. The representations here are shallower than those of the (later) SemEval 2014 shared task, but they were provided for several languages (including Chinese and Spanish). Swayamdipta et al. (CoNLL 2016) presented a transition-based parser that jointly produced syntactic and semantic dependencies. It learns a representation of the entire algorithm state, using stack long short-term memories. Our greedy inference algorithm had linear time, including feature extraction. On the CoNLL 2008–9 English shared tasks, we obtained the best published parsing performance among models that jointly learn syntax and semantics. In particular, we reported a new state of the art for Chinese, and statistically matched the best published results for Spanish.

Abstract meaning representation. Abstract Meaning Representation (AMR) is a semantic formalism for which a growing set of annotated examples is available. Flanigan et al. (ACL 2014) introduced the first approach to parse sentences into this representation, providing a strong baseline for future improvement. The method was based on a novel algorithm for finding a maximum spanning, connected subgraph, embedded within a Lagrangian relaxation of an optimization problem that imposed linguistically inspired constraints. Our approach, known as JAMR, is described in the general framework of structured prediction, allowing future incorporation of additional features and constraints, and may extend to other formalisms as well.

Flanigan et al. (SemEval 2016) presented improvements to the JAMR parser as part of the SemEval 2016 Shared Task 8 on AMR parsing. The major contributions were: improved concept coverage using external resources and features, an improved aligner, and a novel loss function for structured prediction called infinite ramp, which is a generalization of the structured SVM to problems with unreachable training instances.

Liu et al. (NAACL 2015) presented a novel abstractive summarization framework that draws on the AMR treebank. In this framework, the source text was parsed to a set of AMR graphs, the graphs are transformed into a summary graph, and then text was generated from the summary graph. We focused on the graph-to-graph transformation that reduces the source semantic graph into a summary graph, making use of an existing AMR parser and assuming the eventual availability of an AMR-to-text generator. The framework was data-driven, trainable, and not specifically designed for a particular domain. Experiments on gold-standard AMR annotations and system parses showed promising results.

Flanigan et al. (NAACL 2016) turned the challenging task of language generation from purely semantic representations. We addressed generating English from AMR, consisting of re-entrant graphs whose nodes are concepts and edges are relations. The new method was trained statistically from AMR-annotated English and consisted of two major steps: (i) generating an appropriate spanning tree for the AMR, and (ii) applying tree-to-string transducers to generate English. The method relied on discriminative learning and an argument realization model to overcome data sparsity. Initial tests on held-out data showed good promise despite the complexity of the task.

Word vectors. In the span of the DEFT effort, statistical NLP methods, including those designed for semantic analysis, shifted from reliance on carefully-designed features to features automatically discovered using neural networks (also known as "representation learning"). This shift was evident in the project, with earlier work using the former and neural methods prominent in later work (Swayamdipta et al., CoNLL 2016; Peng et al., ACL 2017). At the core of this new class of approaches were distributed, vector space representations of words, to which we made two key advances.

Vector space word representations were learned from distributional information of words in large corpora. Although such statistics are semantically informative, they disregard the valuable information that is contained in semantic lexicons such as WordNet, FrameNet, and the Paraphrase Database. Faruqui et al. (NAACL 2015) proposed a method for refining vector space representations using relational information from semantic lexicons by encouraging linked words to have similar vector representations, and it makes no assumptions about how the input vectors were constructed. Evaluated on a battery of standard lexical semantic evaluation tasks in several languages, we obtained substantial improvements starting with a variety of word vector models. Our refinement method outperformed prior techniques for incorporating semantic lexicons into word vector training algorithms.

Yogatama et al. (ICML 2015) proposed a new method for learning word representations using hierarchical regularization in sparse coding inspired by the linguistic study of word meanings. We showed an efficient learning algorithm based on stochastic proximal methods that is significantly faster than previous approaches, making it possible to perform hierarchical sparse coding on a corpus of billions of word tokens. Experiments on various benchmark tasks—word similarity ranking, syntactic and semantic analogies, sentence completion, and sentiment analysis—demonstrated that the method outperformed or was competitive with state-of-the-art methods.

3.1.2 Some Insights

The lack of consensus on semantic formalisms for NLP, originally viewed as a frustration, is actually a strength. Different attempts to annotate/encode semantics

are complementary but overlapping in the phenomena they capture, and some themes have begun to arise. Better systems result from separating the details of the formalism from the underlying optimization algorithm used to parse, as we can make advances on each separately. We can also exploit heterogeneous datasets together —for example, through multitask learning (Peng et al., ACL 2017) or guide features (Kshirsagar et al., ACL 2015)— to obtain better performance across the board.

Although much of the conversation about symbolic vs. connectionist approaches to language understanding makes the two seem incommensurate, we found that so-called "deep" learning can blend elegantly with the optimization mindset that worked so well in earlier versions of semantic parsing (and linguistic structure prediction more generally). Big gains came from moving from "word vectors" to either "algorithm state vectors" or "part vectors" (or a combination of these) and end-to-end learning.

3.1.3 Data, Code, and Other Products

- Semantic analyzer of frame representations (SEMAFOR)
  Public demo: http://demo.ark.cs.cmu.edu/parse
  https://github.com/Noahs-ARK/semafor
- Supersense-tagged repository of English with a unified semantics for lexical expressions (STREUSLE)
  http://www.cs.cmu.edu/~ark/LexSem
- Multiword expression and supersense tagger (AMALGrAM) and later improvements for SemEval 2014 shared task
  https://github.com/nschneid/pysupersensetagger
- Greedy joint syntactic-semantic parser
  https://github.com/clab/joint-lstm-parser
- Neurboparser for semantic dependencies
  https://github.com/Noahs-ARK/NeurboParser
- JAMR parser and generator for the abstract meaning representation
  https://github.com/jflanigan/jamr
- Summarization based on AMR:
  https://github.com/summarization

3.2 Entity Detection and Linking

Goal: Identify entities in the text and link them into a given database/collection of entities.

Languages: English, Chinese, Spanish

Subproject lead: Eduard Hovy, CMU

Principal participants: Xuezhe Ma, Nicholas Fauceglia, Yiu-Chang Lin, Sujay Jauhar, Evangelia Spiliopoulou, Shuxin Yao (all part of the time)

3.2.1 Final Status of This Subproject

Our trilingual Entity Discovery and Linking module (EDL) was initially developed for the TAC-KBP 2015 EDL track, and since then was extended and improved to address new challenges added for subsequent EDL tracks (Fauceglia et al., 2015; Fauceglia et

al., 2016). In particular, the 2016 and 2017 evaluations 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").

Final performance results for this module appear in Section 4.1.1.

Our end-to-end EDL system included XML document file parsing, entity extraction, linking, type inference and NIL clustering. We used the Stanford CoreNLP pipeline (Manning et al., 2014) for preprocessing and named entity recognition, and adapted and extended (Moro et al., 2014) for entity extraction and linking. Our system first processed all of Wikipedia, representing it as a directed weighted graph, and then computed a semantic signature for each vertex. Second, we used these semantic signatures for entity discovery and linking across three languages in a system that used an extended version of Babelfy (http://babelfy.org) as its backbone.

We used Babelfy (http://babelfy.org) as the backbone of our system and extended/adapted it to be suited for the EDL task. Our system differs from Babelfy in the following points:

- The system used Wikipedia's Ontology directly, instead of merging WordNet into the KB.
- For the construction of semantic signature, we used the algorithm Personalized PageRank with node-dependent restart (Avrachenkov et al. 2014), instead of Random Walk with Restart (Tong et al. 2006).
- We modified the candidate extraction method and extend it to Chinese and Spanish.
- We introduced edge weights to semantic interpretation graph.
- We proposed a new rule-based entity type inference method.
- We trained a joint word and entity embeddings to handle nominal mentions.

Our development work from 2015–17 resulted in significant improvement on all three languages. We briefly describe the main processing steps below and evaluation results in Section 4.1.1; details can be found in (Ma t al., 2017).

Our system first constructed 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) to which this edge belongs. To implement the graph, we used the WebGraph framework (Boldi and Vigna, 2004).

Using the completed Wikipedia graph, we computed a *semantic signature* for each vertex, namely the set of other vertices strongly related to it. To calculate semantic

signatures, we first computed the transition probability $P(v'|v)$ as the normalized weight of the edge:

$$P(v'|v) = w(v|v') \;/\; \Sigma_{v'' \in V}\, w(v|v'')$$

Where $w(v'|v)$ is the weight of the edge v —> v'. With the transition probabilities, Semantic Signatures were computed using Personalized PageRank with node-dependent restart (Avrachenkov et al., 2014), which differs from the Babelfy approach. Vertices with a score lower than a threshold were discarded. Different from previous versions of SAFT, we eventually 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 performed *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 focused on named mentions within a certain window size around the nominal and exploited 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* was constructed by uniting all semantic signatures of every candidate. A graph densification algorithm was then applied iteratively until a density threshold was reached: at each step of graph densification, we first found the most ambiguous mention (the one with the most candidate entities), then removed the candidate entity from the most ambiguous mention that has the smallest score. The score of vertex (v, f) in the semantic interpretation graph differed from the one in Babelfy—we used the sum of the incoming and outgoing edge *weights* instead of the sum of incoming and outgoing degrees. Formally, the score of the vertex (v, f) is:

$$score((v, f)) = w(v, f) \cdot sum((v, f)) \;/\; \Sigma_{(v', f)}\, w(v', f) \cdot sum((v', f))$$

Where $sum((v, f))$ is the sum of the incoming and outgoing edge weights of (v, f) and $w((v, f))$ is the number of fragments the candidate entity v connects to. The above steps were repeated until every mention had less than a certain number μ of candidate entities. Finally, we linked each mention f to the highest-ranking candidate entity v^* (including NIL) if its $score((v^*, f)) > \theta$, where θ was a fixed threshold.

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

3.2.2 Advances Made in This Subproject

We built the first EDL system at CMU. It has been reused and extended in subsequent projects, including the ARIEL project in DARPA's Low Resource Languages for Emergent Incidents (LORELEI) program and the OPERA project in DARPA's Active Interpretation of Disparate Alternatives (AIDA) program.

3.3 Events: Mention (Nugget) Detection, Coreference, Script Structure

Goal: Identify event mentions and relate fully and partially identical events and entities.
Languages: English and Chinese
Subproject lead: Teruko Mitamura and Eduard Hovy, CMU
Principal participants: Jun Araki (partially), Hector Zhengzhong Liu

3.3.1 Final Status of This Subproject

Semantics-oriented processing of events was a strong novel focus throughout the years of the DEFT program. This work was on the forefront of it and helped significantly drive community discussion, task breakdowns, and evaluation decisions.

The software built here included several modules, addressing different aspects of the event processing tasks. Final performance results for these modules are presented in Section 4.1.1.

Early KBP tasks focused exclusively on the extraction of entities and slot relations (e.g., *per:spouse*). But a major goal of the 2017 Cold Start++ task was the integration of event information, and this is what we focus on in this report. Some of the event types shown in Figure 1. The following aspects of events were 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.

Through the years, all these aspects were evaluated in standalone event evaluations, and their integration into Cold Start++ followed the guidelines from those standalone evaluations. The only exception was cross-document event coreference, which was 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 had a rich portfolio of event extraction systems built up over successive TAC-KBP event evaluations, which were further extended and refined for Cold Start++. Those systems are labeled A-to-C in Figure 3. In addition, a new

system D was developed in 2017 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).

Event nugget detection and Coreference module A: The goal of this module was to detect event nugget instances and coreference clusters that group together the nuggets referring to the same underlying events. The targeted languages were English and Chinese. The nugget detector was developed over several TAC-KBP evaluations (Liu et al., 2015; Liu et al., 2016). It employed a Conditional Random Field (CRF) model (Lafferty et al., 2001) trained discriminatively using the Passive-Aggressive algorithm. We used a number of other tools for preprocessing, syntactic as well as semantic parsing: CoreNLP, SEMAFOR (Das et al., 2014), FANSE Parser (http://www.isi.edu/publications/licensed-sw/fanseparser/) as well as the LTP (http://ltp.ai) parser and toolkit for Chinese. We first used the semantic parser SEMAFOR (Section 3.1) to generate a set of candidate Event Nuggets, their FrameNet frame, and their Frame Elements. We then used the Stanford CoreNLP POS tagger (Manning et al. 2014) to classify the candidate events as verbal and nominal events. For every trigger in the candidate events, we used the output frame to decide whether it was an event or not. If the frame was in the domain of the FrameNet-to-ACE mapping, then it corresponded to some subtype of the ACE Ontology and we accepted it as an event. Using the mapping we assigned the type and subtype of the event. Figure 1 shows an example output of the system for one article.

Type	Subtype	Event Nugget
Life	Die	killed
Life	Injure	wounded
Conflict	Attack	blast
Life	Die	death
Contact	Phone-Write	Radio
Life	Die	death
Transaction	Transfer-Money	giving
Life	Die	deaths
Life	Injure	wounded
Life	Injure	injured
Contact	Phone-Write	list
Conflict	Attack	bomb
Conflict	Attack	tore
Conflict	Attack	explosion
Conflict	Attack	hit
Life	Injure	injuries
Conflict	Attack	blast
Conflict	Attack	bomb
Conflict	Attack	exploded
Contact	Phone-Write	radio
Movement	Transport	carting
Life	Injure	wounded

Figure 1: Example of Event and Event Nugget Output

The events are represented with green, red and black color if they are true positives, false positives and false negatives, respectively.

The following traditional linguistic features were 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 included 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).

Event Coreference Resolution systems have a structure similar to Entity Coreference systems. Both normally comprise a mention detection step and a clustering step. While joint learning of the two steps is successful for entity coreference, it remained a question whether similar techniques could be applied to event coreference. We experimented with a simple Dual Decomposition-based joint decoding method to jointly perform these two tasks. Experiments on the semantic oriented TAC-KBP corpus showed that the joint decoding method gave a small but consistent improvement. While given mention spans, our joint decoding method improved all the coreference metric by a small margin.

On analyzing the reason for the small gain, we found that deep semantic understanding is required to solve the problem. Event coreference is highly affected by Event Type Detection. Since our early event coreference algorithm was almost entirely determined by event type matching, we saw a relatively small improvement when using the joint decoding model. To make the Event Coreference component less reliant on type detection, we needed to solve some deep semantic inference problems. For example, one needs to be able to solve ambiguous argument reference (e.g., last Wednesday vs. last week), which goes beyond simple entity coreference. Such phenomena can be seen in the diagram below, while resolving coreference is difficult through other means, simply merging based on event types gave almost perfect output.

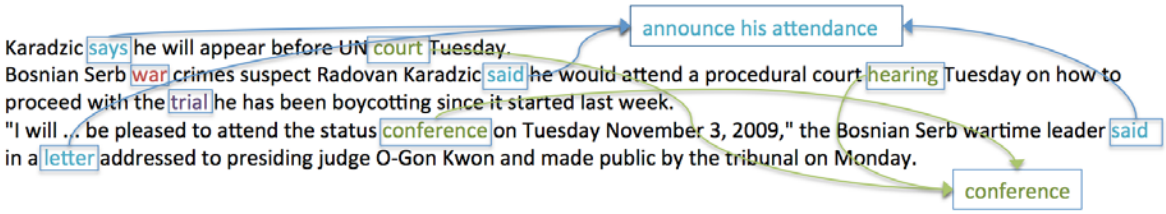

Figure 2: Example of Event Hopper Annotations

Figure 2 shows an example of Event Hopper Annotations (all mentions of the same type are coreferent in this snippet). The structural semantics of event mentions, however, is very difficult to extract and to compare, hence these normally bring little benefits in our learning framework.
Our coreference resolver was a Latent Antecedent Tree model that constructed 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 arguments
- Discourse features: sentence and mention distances

Matching between coreference candidates was based on word vector similarity, Brown cluster matching, WordNet sense matching, POS, lemma, mention type and mention realis. Event nugget and coreference results were then merged with those from other CMU SAFT team systems to create a merged output (see Section 4.1.2).

Event nugget detection and coreference module B: Similar to module A, the goal of module B was to identify event spans in text and assign an event type to each detected span. The targeted languages were English and Spanish. Module B could 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 formalized 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 was 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 was an extension of a neural event detection system developed for TAC-KBP 2016 (Liu et al., 2016) that used 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 dependencies between labels in neighborhoods and jointly decode the best sequence of labels for an in- put sentence. Therefore, instead of decoding labels independently, we modeled 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 used the TAC KBP 2015 event nugget dataset. We also employed pre-trained 50-dimensional GloVe word vectors (Pennington et al., 2014) and did not fine-tune them during training. We used Adam (Kingma and Ba, 2015) to optimize parameters based on the performance in the validation dataset. Our experiments showed that the model achieved 61.90 F1 in span detection and 55.91 F1 in span+type detection on the TAC KBP 2015 test data (English). This performance was close to the state-of-the-art, and it was ranked third in the official results of TAC KBP 2015.

3.3.2 Advances Made in This Subproject

We built several event mention detection and coreference systems. They are being reused and extended in subsequent projects, including the OPERA project in DARPA's AIDA program. Some systems have been uploaded to GitHub for distribution.

3.4 Event Relation Extraction

Goal: Identify the principal event(s) in a document and extract its/their participants.
Languages: English, Chinese, Spanish
Subproject lead: Module C: Jaime Carbonell and Yiming Yang, CMU; Module D: Eduard Hovy, CMU
Principal participants: Module C: Andrew Hsi; Module D: Evangelia Spiliopoulou, CMU

3.4.1 Final Status of This Subproject

The primary goal of these modules was to extract event arguments for each event discovered by the various SAFT event nugget modules. The targeted languages were English, Spanish and Chinese. Our software includes several modules. Final performance results for these modules are presented in Section 4.1.1.

Event nugget detection and participant relation detection module C: This module 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 began by performing preprocessing using Stanford CoreNLP and the MaltParser (http://www.maltparser.org/) on the input documents to extract information such as tokenization, part of speech tags and dependency parses. We then obtained 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 3.2. We then obtained event

nugget information from (1) the CRF- based event nugget detection module A described in Section 3.3 designed for English and Chinese, (2) the BiLSTM-CRF-based event nugget detection module B also described in Section 3.3 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 3).

The output from entity extraction and nugget detection was then fed into a logistic regression argument classifier, which made predictions of argument relationships on every entity/nugget word pair within the same sentence. Finally, a realis label was predicted for each discovered argument, once again using logistic regression. For training, we used the ACE 2005 and Rich ERE datasets. Word embeddings for all three languages were obtained from their respective Wikipedia dumps using word2vec. Arguments were extracted separately for each set of event nuggets coming from modules A, B and C and were then merged by the Event Merger (see Section 4.1.2).

Our logistic regression classifiers used 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 was most noticeable when there is little annotated event training data available (as is the case for Spanish).

One special feature of this work was cross-lingual training to create a single, cross-lingual model, rather than separate models for each language. This was motivated by previous success in the NLP literature for cross-lingual applications (Richman and Schone 2008; Zeman and Resnik 2008; Snyder et al. 2009; Chen and Ji 2009; Cohen et al. 2011; McDonald et al. 2011; Piskorski et al. 2011; Ammar et al. 2016; Hsi et al. 2016b). Such models can be particularly useful when there is little training data available for a particular language (as is the case for Spanish event extraction), but much more data available for other resource-rich languages (e.g., English). This model used a combination of language-dependent and language-independent features, which allowed the model to capture general patterns across languages as well as specific nuances found in individual languages. Our language-independent features covered information obtained by Universal POS tags (Petrov et al. 2012), nugget type information, entity type information, and Universal Dependencies (McDonald et al. 2013), while our language-dependent features included information based on individual words, language-specific part-of-speech tags, and word embeddings. The overall model was then trained by simply using all available annotated data (across all three languages) at training time.

Event nugget detection and participant relation detection module D: This module is described in more detail in (Spiliopoulou et al., 2017); here we just briefly list its main characteristics. The goal of this module was again event nugget

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

The main idea behind our event detection approach was that frame-semantic parsers generate a rich set of predicates that can directly serve as event nuggets. To this end, our approach started with the output of a frame-semantic parser that was then refined in order to get a large set of event nugget candidates. This allowed us to exploit the rich semantic structure generated by such a parser to generate more event candidates and achieve higher recall than previous systems.

In order to generate a list of candidate events, we used SEMAFOR (Das et al., 2014), described in Section 3.1, which is a frame-semantic parser based on FrameNet (Fillmore et al., 2003). 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 utilized structural similarities between FrameNet and the TAC KBP ontology. We observed that most types of the TAC KBP event ontology could be decomposed into a small set of FrameNet frames. Thus, we first manually constructed 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 detected our event nuggets based on this mapping: a mention generated by SEMAFOR was accepted as an event only if its frame is in the domain of the mapping.

The final part of the system involved 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 used 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 assigned a named entity as time or location of an event nugget only if both occur in the same sentence.

3.4.2 Advances Made in This Subproject

We built two event participant detection systems. Their final performance results are reported in Section 4.1.1.

3.5 Inference/Relation Discovery

Goal: Automatically discover inference rules by analyzing patterns and correspondences in text.
Languages: English
Subproject lead: William Cohen, CMU
Principal participants: William Yang Wang and programmer Kathryn Mataitis (partial support)

3.5.1 Final Status of This Subproject

One of the major challenges in DEFT was knowledge base completion. The motivation is clear: given a partially complete knowledge base, how can we utilize background knowledge and additional information to automatically reason to generate new facts? In this context, the goal of this subproject focused on two aspects. First, we were interested in leveraging context information for knowledge base completion. In prior work, most of the studies focused only on relational triples, ignoring the importance of text-based evidence. On the other hand, we were also interested in improving information extraction (IE) by considering joint IE and relational reasoning, since most of the relational extractors did not perform relational inference during the extraction step.

We developed an alternative strategy for knowledge-base completion (KBC) based on a hybrid neural/logical model, in which probabilistic logical formulae in ProPPR were embedded into real vectors, and matrix factorization methods were used to generalize a matrix of indexed by potential inferences (as rows) and supporting formulae (as columns). This method outperformed the state-of-the-art in KBC at the time of its publication (in IJCAI 2016).

We prototyped a successor to the ProPPR probabilistic reasoning system called TensorLog which is more compatible with gradient-based learning methods and deep learners. This required many fundamental changes, including a new and different inference strategy.

We evaluated the ProPPR-based approach on the full KB produced by SAFT on the TREC/KBP task. We were able to obtain high-quality (MAP ~= 0.9) inferences for 27 relations. For the remaining three relations (*person:title, type, person:alternate_names*) essentially no inference rules were learned.
TensorLog was extended to compile to existing neural infrastructures, notably Tensorflow, which gives it much better performance in certain settings. In particular, TensorLog is better suited to problems with many labeled examples.
A novel structure-learning extension to TensorLog was developed and evaluated on the SAFT KB. Although this method outperformed the ProPPR approach on benchmark datasets, its performance on the SAFT KB was slightly worse than ProPPR. We evaluated and delivered the ProPPR-based approach for KBC with the infrastructure needed to test and evaluate it.

3.5.2 Advances Made in This Subproject

We designed and tested several inference discovery algorithms, as described above. These were not however used in the overall system assembly or evaluations.
We delivered the following code:

- Regression testing infrastructure for KBC:
  - RuleLearnerRegressionTest, RuleLearnerProfiler, RuleLearnerBenchmarkTest, KB adapter for module unit tests in Adept (RegressionTest, Profiler, and BenchmarkTest)

- Machinery for testing KB modules on the Adept KB in a read-only fashion
- Machinery for verifying facts were added to the KB during unit tests (specifically facts without document support)
- ProPPR integration with Maven

- Generalized ProPPR structure learning pipeline to support both validation (split/train/predict/eval) and inference (train/predict)
  - Reservoir sampling of the KB for train/test split
  - Inheritance-aware, typesafe query generator for inference mode
  - Inheritance-aware, typesafe prediction filter for promoting new facts into the KB
  - Adapters for handling order-based arguments (ProPPR) vs role-based arguments (Adept)
  - Converters for existing ProPPR benchmark to work with the hard-coded Adept ontology

3.6 System Integration

Goal: Integrate the results of all the above subsystems and produce a single Cold Start-style knowledge base for evaluation in the KBP challenge.
Subproject lead: Hans Chalupsky, USC Information Sciences Institute, Marina del Rey, CA
Principal participants: Jun Araki, Eduard Hovy, Andrew Hsi, Hector Zhengzhong Liu, Xuezhe Ma, Evangelia Spiliopoulou, and Shuxin Yao

3.6.1 Final Status of This Subproject

This section summarizes the system integration work by focusing on our participation in the TAC-KBP 2017 Cold Start++ Knowledge Base Population task. Our submitted SAFT system was a loosely-coupled integration of individual components as described in previous sections that processed documents in English, Spanish, and Chinese. The system extracted entities, slot relations, event nuggets and arguments, performs entity linking against Freebase, event coreference, and 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 significant extension of the Cold Start KBP (CS-KBP) tasks held in previous years along a number of dimensions:

(1) It built upon previous SAFT work in all three languages (English, Spanish and Chinese). While the 2016 CS-KBP task was also organized for all three languages, participants were free to choose which language conditions to participate in.
(2) It fully integrated the entity discovery and linking aspect, which had previously been evaluated separately. A CS-KBP 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) It fully integrated a tri-lingual version of the event nugget detection and linking tasks held in previous years.
(4) It included the tri-lingual version of the event argument detection and linking tasks from 2015 and 2016.
(5) It included a tri-lingual sentiment detection task (somewhat simplified relative to the full BeSt tasks organized in prior years and focusing only on sentiment relations between person entities).

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 Chinese slot filling relations where we finished and ran the extractor but failed to integrate its results, and 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.

3.6.2 SAFT Cold Start++ System Architecture

Figure 3 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 de- fined 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).

Our team members ran their individual modules on different machines, and upon completion result files were archived and shipped to other team members to allow their modules to run using those results as inputs. Each component processed its inputs fully automatically, some then sent their results automatically to downstream components (e.g., from English Slot Relations to KResolver Mini-KB Integration), while others required 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 in Figure 3 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.

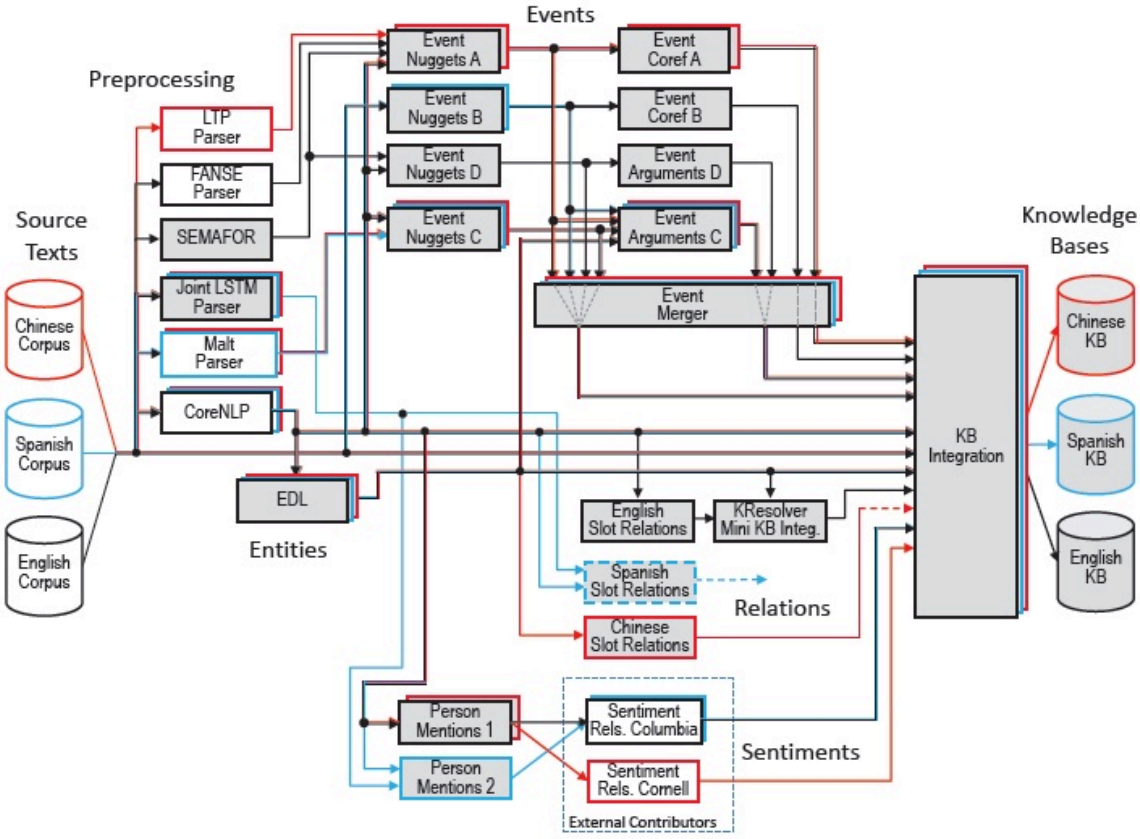

Figure 3: Final SAFT architecture

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 exceptions were 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 pre-processing had been finished, but we failed to complete and run the extractor on the Spanish document set. We provided detail on individual component modules and their respective performance in previous sections.

These were evaluated using four kinds of evaluation metrics: (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. Evaluation results presented in section 4.

3.7 Evaluation Organization: TAC Event Mention Detection and Coreference Track

During the duration of the DEFT program, Mitamura and Hovy defined, set up, and helped organize the TAC 2017 Event Mention Detection and Coreference evaluation tasks, which were annually administered by NIST and reported at the annual TAC conferences every November. The website for the 2017 task is at http://www.nist.gov/tac/2017/KBP/Event/index.html

Tasks: The three tasks we were responsible for are:

- **Task 1: Event Nugget Detection**: This task aimed to identify the explicit mentioning of Events in text for English. Participating systems had to identify all relevant Event Mention instances within each sentence. Every instance of a mention of the relevant Event types/subtypes taken from the Rich ERE Annotation guidelines had to be identified. In addition, systems had to identify three REALIS values (ACTUAL, GENERIC, OTHER), which were described in the Rich ERE guidelines.
- **Task 2: Event Nugget Detection and Coreference**: In addition to the Event Nugget Detection task described in Task 1, this task aimed to identify Full Event Coreference links at the same time. Full Event Coreference was identified when two or more Event Nuggets referred to the same event. This notion is described as *Event Hoppers* in the Rich ERE Annotation Guidelines. The Full Event Coreference links did not include subevent relations.
- **Task 3: Event Nugget Coreference**: This task was to identify Full Event Coreference links, given the annotated Event Nuggets in the text.

Scorer: We built an automated scorer, which was updated and used in the evaluations. It is available at https://github.com/hunterhector/EvmEval.

3.8 The EVENTS Workshop Series

Mitamura, Hovy, and students co-organized an annual series of workshops dedicated to the processing of events, called the EVENTS workshops. Initially these workshops were held in the US with the NAACL HLT conferences, but in 2016 and 2017 we combined them with the News Stories series, organized primarily out of the EU, to form the international EventStory Workshops. The most recent one-day workshop was held at the ACL conference in Vancouver on August 4, 2017. As invited speakers we were fortunate every year to secure a very illustrious speaker, including Prof. James Pustejovsky (Brandeis University), Dr. Jerry Hobbs (USC/ISI), Dr. Bernardo Magnini (FBK IRST, Italy), Prof. James Allen of IHMC / University of

Rochester. Each workshop included at least a dozen papers (some of them oral) from the USA, Canada, Japan, Norway, Germany, EU, The Netherlands, and elsewhere, covering many aspects of events (including ontologies, corpora, storylines, causality, temporal linking, and events in Twitter streams). Typically, more than 50 participants attended, reflecting the growing international interest in working on events.

4 Results and Discussions

4.1 Modules' Performance in the 2017 KBP Task

In this section we review the performance of the various modules in the 2017 ColdStart KBP task.

4.1.1 Entity Discovery and Linking Results

Our trilingual Entity Discovery and Linking module (EDL) is described in Section 3.2 above. 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. In Table 2 there are no results for Spanish and Chinese because our system only handled English nominal mentions.

Table 1 Standalone TEDL evaluation on NER, Linking, and Clustering.

	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 2. Standalone TEDL evaluation on Named and Nominal Mentions

	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	–	–	–	–	–	–	–	–	–

4.1.2 Event related Results

Event Extraction and Coreference: Our event mention detection system was described in Section 3.3 above. Due to the merging of multiple event module outputs, we do not have individual results for this module alone. In 2017 our component-level results (detailed in Table 7) were F1=35.85 mention span detection score for English, which was the top result for all Cold Start++ teams in 2017. Our coreference score that year was 13.56 in terms of KBP average, which ranked at second place for English. Our Chinese systems performance was slightly lower at F1=29.06, and the coreference KBP average was 8.71. Note that most individual KBP participants had a higher recall than precision, while we observed the opposite. This was likely due to the fact that we merged the results of multiple systems.

Event Nugget Detection and Participant Arguments module C:
The primary goal of this module was to extract event arguments for each event discovered by the various SAFT event nugget modules. The module was described in Section 3.4 above. Table 3 summarizes the evaluation results for this component on the TAC-KBP 2017 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. standalone event argument extraction task. Details of our participation and results are described in (Hsi et al., 2017).

Table 3. Standalone event argument and linking results for Event Arguments

	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

Event Nugget Detection and Participant Arguments module D: The primary goal of this module was to both identify event nuggets and for them extract event arguments for each event discovered by the various SAFT event nugget modules. The module was also described in Section 3.4.

Event Merging. This module combined the outputs of the four event detection systems described immediately above. It generated a set of merged event nuggets for all three languages, a set of merged event arguments for all three languages, and integrated the event coreference clusters from modules A and B into a uniform format but left them unmerged.

The module first took the union of the collected outputs (nuggets and arguments) and then applied a neural net classifier to provide confidence scores for each event nugget instance. These confidence scores depended only on the type of event that each system predicted for every candidate event (which is *None* if a system did not classify a certain mention as an event). We did not develop a mechanism to compute confidence values for event arguments.

4.1.3 Slot Filling Results

Slot Relation Extraction. The relation extraction systems shown in the lower part of Figure 3 extracted the 65 TAC-KBP Cold Start slot relations used in 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 were relatively recent additions to the task, was 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.

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

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

KRes is a logic-based inference system based on the PowerLoom knowledge representation and reasoning system (http://www.isi.edu/isd/LOOM/PowerLoom/) 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 Power- Loom 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 was an extension of previous versions that (1) was extended to address the full set of TAC-KBP slots, (2) used a single multi-class classifier instead of the binary classifiers used before, (3) did not use features based on SEMAFOR, and (4) added some new features such as Brown clusters compared to what is described in (Chalupsky, 2014).

The extractor started by detecting possible arguments of types relevant to TAC-KBP slots. Argument mentions and their types were constructed from NER-types detected by CoreNLP, Wordnet, as well as gazetteers such as title lists. It then enumerated possible argument pairs within a certain maximum distance in the dependency tree and then classified each pair using a 30-class maximum entropy classifier. To keep the set of classes as small as possible we normalized each relation onto its canonical forward form and combined the various city/state/country slots onto place slots such as *place of residence*, which were 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 was more or less identical to previous versions and is described in more detail in (Chalupsky, 2013).

The result of this extraction process was a set of typed, sentence-level relation mentions whose arguments might be named mentions, nominals, pronouns or values such as ages. Additionally, we had 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. See below for elevant composite and component evaluation results for English slot relations.

KResolver Mini-KB Integration. The second phase of English slot relation extraction was what we call Mini-KB generation, which produced consistent per-document KBs in TAC-KBP format for each document in the corpus. These document-level mini-KBs were then combined into a global raw KB, which was then further refined and de-conflicted (see Section 4.1.2).

The main advantage of this scheme was scalability, since it allowed us to use more expensive inferencing on a smaller, focused, per-document basis, which in addition could be performed in parallel, since documents could be processed independently. The disadvantage was that it prevented us from performing more fine-grained adjudication of conflicts when looking across documents.
The Mini-KB integration phase took 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 linked equivalent entity mentions into entities, and equivalent relation mentions into relations, which then formed an initial raw knowledge base. The challenge was that all mention-level information is noisy, incomplete, redundant, fully or partially overlapping, and possibly inconsistent. In particular, once equivalences were introduced, type information from equivalent mentions started 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* be- tween "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 were treated as separate *hypotheses*. Specifically we generated *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 mapped 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 generated 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 derived its narrower type from the corresponding Freebase entry. For unlinked mentions, we used a set of city, state and country gazetteers derived from the GeoNames (www.geonames.org) database.

In the linking phase we performed incremental *what-if* analyses of subsets of these hypotheses to see which combinations lead to conflicts and what the culprits of these conflicts were. This part of the system heavily leveraged PowerLoom's multi-context reasoning as well as its explanation system. We used a greedy scheme that started by asserting all type and relation hypotheses in a hypothetical reasoning context. We then queried for all type and constraint violations, and for each violation found we analyzed the proof tree to find the set of extraction hypotheses underlying the conflict. This allowed 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 retracted the weakest hypothesis in a conflict support to remedy the conflict. Hypothesis strengths were based on classifier confidences or heuristics where those were not available (e.g., relation hypotheses are generally weaker than entity type hypotheses). Next we asserted mention overlap equivalence hypotheses and repeated this process, then the same for name links, coref links implied by EDL and then general coref links from CoreNLP. This process introduced noisier and noisier information at each stage and then retracted the weakest hypotheses underlying any newly discovered conflicts. At the end we used the set of surviving type, relation and equivalence hypotheses to form the mini-KB for the current document.

Finally, we translated 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 translated into *per:city of death*, too fine-grained family relation such as *hasNiece* were mapped onto *per:other* family, and we also performed some other inferences for inverse slots and inferring employment from top employment. We additionally performed 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 did not yet eliminate redundancies; this was left to Phase 3 of the KB construction process (see Section 4.1.2). Most importantly, entities linked to Freebase m-IDs or NIL clusters by EDL received KB IDs based on these identifiers, which automatically linked them with corresponding entities from other documents, thus, forming a globally linked knowledge base.

Spanish Slots: Participant Relations. The Spanish Slot Relations module's primary goal was to extract the 65 CS-KBP slot relations from Spanish documents. Given the relatively short amount of time and limited manpower 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 had 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 are 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 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 (http://crscardellino.me/SBWCE/) 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 3 is drawn with dotted lines and we do not have any Spanish evaluation results for queries that involved any slot relations.

Chinese Slots: Participant Relations. Our Chinese relation extraction module aimed to find CSKB slot relations from Chinese text. Slot relations comprised 41 base relations and their inverses. In order to automatically extract slot relations,

we used an ensemble of rule-based classifiers and a bidirectional 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 contains 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 followed the idea of (Lin et al., 2016), (Zhou et al., 2016) and the work from the Natural Language Processing Lab at Tsinghua University (https://github.com/thunlp/TensorFlow-NRE) and used a bidirectional 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 started by constructing a Chinese character embedding using a skip-gram model (Mikolov et al., 2013). All words with fewer than 5 occurrences were removed, numbers and dates were replaced with special tokens, and named entities were recognized and concatenated together by underscores.
For each training instance to the neural relation extraction model we had 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 was 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 that 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 was used to encode each sentence. The hidden representations of each time step from both directions were concatenated as the features of each word. We used two attention mechanisms on different granularities: word level and sentence level. Word-level attention calculated 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 was used to represent the sentence. Sentence-level attention calculated 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 calculated the probability of each relation as well as cross-entropy for calculating loss.

Our approach formed a corpus-level relation extractor that predicted relations between entity pairs collectively based on all sentences in the corpus where two entities co-occur. This was different from more traditional sentence-based approaches as used, for example, by our English slot relations component. Since a Cold Start++ submission required 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 was not 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 4 shows total Area Under the Curve (AUC) as well as precision numbers at the top-scoring 300, 600, and 900 relation instances.

Table 4. **Results of Chinese slot relation extraction (internal evaluation)**

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

4.1.4 Sentiment related Results

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 took document annotations in LDC's ERE format as input and produced sentiment

annotations in the BeSt XML format developed during past TAC-KBP belief and sentiment evaluations.

The trilingual sentiment detection task of Cold Start++ 2017 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++ 2017, sentiments were therefore represented only by two relation, person-to-person *per:likes* and *per:dislikes* 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.

4.2 Knowledge Base Integration

The last box in Figure 3 is the KB Integration component. It took all outputs from any extraction component across all three languages in addition to source texts and CoreNLP annotations and produced one KB file per language.

In SAFT, 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 4.1.1, once extractions were combined across components and across documents, conflicts could arise which led to an inconsistent knowledge base. These conflicts had 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 was that it was an inherently global task that needed to take into account all or large portions of the entire corpus data at once. For this reason, it was 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 4.1.1.

Our basic approach to KB integration was as follows: we started by building a raw KB that unions and links outputs from individual components by introducing, normalizing, and merging KB node IDs as necessary. We started with EDL output, which was taken more or less literally and only reformatted to conform to the Cold Start++ output format. We did not have meaningful confidences for EDL mentions, so all mentions added to the KB received a confidence score of 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 required 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 that 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 produced 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 was fairly straightforward. All that had to be done here was to link them to other nuggets or events from event arguments via their within-document event coreference links. We again used merged type, realis, and confidence provided by the Event Merger.

Finally, we output sentiment relations. Similar to the event arguments case, we tried to match sentiment relation source and target mentions to entities introduced by EDL or KRes. If no match could be found, we introduced new document-local entities for source and/or target. We used 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 had all the necessary information to output an initial raw knowledge base. We built this raw KB by combining all available linked information with associated provenance, and then performed some initial per-document canonicalization which mapped all inverse relations onto their corresponding forward relations, then eliminated all (now redundant) inverse slots, then removed document-level duplicates and finally added canonical mentions for each entity in a document. At this point, however, this raw KB contained 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 counted the number of explicit and implicit type judgments for each entity in the raw KB. Explicit types came from EDL and/or the KRes mini-KBs and are counted once per document. Implicit judgments came from unique domain or range types of slot, event and sentiment relations an entity participates in, and were counted once per mention. We then found problem entities with more than one type and picked a preferred type based on the counts computed before. All KB type assertions and relations that conflicted with this preferred type were 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++ 2017, however, we retained low-confidence results to try to boost recall, which led to some unexpected and undiscovered problems discussed in the evaluation section below.

Finally, additional KB-level refinements had to 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.

4.3 Full System 2017 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 Free- base 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. The Cold Start++ 2017 evaluation was designed to also allow for component-based evaluations for EDL, event argument, nugget and sentiment components, based on a subset of 500 core documents for which a comprehensive gold standard had been created by LDC. This gold standard contained 167 English, 166 Spanish and 167 Chinese documents. We first describe 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++ 2017 evaluation.

Table 5. The official 2017 Cold Start++ KB EDL component results

	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 summarizes official EDL component results based on our KB submissions for all three languages for three key metrics: strong typed mention match (NER), strong typed all match (Linking), and mention CEAF (Clustering). 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 provided 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 2017 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 did 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 were 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 did not have conflicting types from other components that it could be confused with. Only when a correct event argument was identified with an EDL entity whose type was changed to something incorrect did we lose a correct event result due to our deconfliction problem. This is a much more uncommon situation which explains why the effects on the overall argument detection F1 were 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 the 2017 evaluation.

Table 6. The official 2017 Cold Start++ KB event argument component results

	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 7. The official 2017 Cold Start++ KB event nugget component

	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

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 picked 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 in 2017, which took result confidences into account, and were therefore not directly comparable to results from previous years. This also meant that results were 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, since 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 were generally very low, since they always require a correct intermediate entity. For this reason, we primarily discuss Hop-0 results here. As described earlier, our event components were the most mature which is apparent in both the English and Spanish submissions. We remain unsure why event-only queries all failed for Chinese despite the fact that our event argument results for Chinese were actually the best for all three languages. For English, precision significantly dominated recall for all slot dimensions. For Spanish, results were generally very low and reflect the lack of resources and training data. For Chinese, somewhat unexpectedly, sentiment results were quite good and the best of all component-level results for all three languages.

Table 8. The official 2017 Cold Start++ KB query-based composite results

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

5 Conclusions

As the final project report, we here described the SAFT project's various groups and associated modules, and their integration into the end-to-end SAFT system and its participation in the TAC-KBP 2017 Cold Start++ Knowledge Base Population task.

This discussion well summarizes the work we performed in previous years to build all the components. Our system performed well for event nuggets, respectably for event arguments and entity discovery and linking, but relatively poorly for slot relations and overall composite query-based evaluation of the resulting KBs. We observe that NLP pipelines are generally very complex, and that multilinguality and the focus on multiple target modalities exponentiate this complexity. Our hope that combining entity, event, and relation extractions would provide redundancies that would improve overall results (at least in some areas) was not realized, partly due to the immaturity of our integration components. One key insight from all our work in SAFT it is that robust integration of multi-component NLP extractions for KB generation is itself a formidable challenge requiring significant research beyond traditional NLP research itself.

The implemented algorithmic results of our work have been uploaded to the system integrator's site hosted by BBN and through which it was made available to DARPA. Each group has submitted at least one version of its code. This included:

- SEMAFOR and related systems; see Section 3.1 (Smith)
- Entity Detection and Linking module; see Section 3.2 (Hovy)
- Event coreference resolution module; see Section 3.3 module A (Hovy and Mitamura)
- Multilingual event relation extraction module; see Section 3.4 module C (Carbonell and Yang)
- Regression testing infrastructure for KBC; and generalized ProPPR structure learning pipeline to support both validation and inference; see Section 3.5 (Cohen)
- KBP inference module; see Section 3.6 (Chalupsky)

6 Recommendations

None

7 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 Com- putational 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 *Proceedings 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.

Appendix: Project Publications

Not all work published here was entirely funded by this project.

Ammar, W., George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Many languages, one parser. *Transactions of the ACL*.

Araki, J., Eduard Hovy, and Teruko Mitamura. 2014a. Evaluation for Partial Event Coreference. *Proceedings of ACL 2014 Workshop on Events: Definition, Detection, Coreference, and Representation,* p 68.

Araki, J., Zhengzhong Liu, Eduard Hovy, and Teruko Mitamura. 2014b. Detecting Subevent Structure for Event Coreference Resolution. *Proceedings of Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC).*

Araki, J. and T. Mitamura. 2015. Joint Event Trigger Identification and Event Coreference Resolution with Structured Perceptron. *Proceedings of the EMNLP conference.* Lisbon, Portugal.

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

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

Chalupsky, H. 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.

Chen, Y.-N., W.Y. Wang, and A.J. Rudnicky. 2013. Unsupervised Induction and Filling of Semantic Slots for Spoken Dialogue Systems using Frame-Semantic Parsing. *Proceedings of the ASRU conference.* **Best Paper Award**.

Cohen, W.W. and Fan Yang. 2017. TensorLog: Deep Learning Meets Probabilistic Databases. Under journal review; reprint published in arxiv.org 1707.05390.

Cohen, S.B., Dipanjan Das, and Noah A. Smith. 2011. Unsupervised structure prediction with non-parallel multilingual guidance. *Proceedings of the EMNLP conference*.

Dai, Z., A. Almahairi, P. Bachman, E.H. Hovy, and A. Courville. 2017. Calibrating Energy-Based Generative Adversarial Networks. Poster, in *Proceedings of the ICLR conference*. Toulon, France.

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

Dasigi, P. and Eduard Hovy. 2014. Modeling Newswire Events using Neural Networks for Anomaly Detection. *Proceedings of the COLING conference*.

Dasigi, P., W. Ammar, C. Dyer, and E.H. Hovy. 2017. Ontology-Aware Token Embeddings for Prepositional Phrase Attachment. *Proceedings of the ACL conference*. Vancouver, Canada.

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

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

Flanigan, J., Sam Thomson, Jaime Carbonell, Chris Dyer and Noah A. Smith. A Discriminative Graph-Based Parser for the Abstract Meaning Representation. *Proceedings of ACL conference*. code: https://github.com/jflanigan/jamr

Goyal, K., S.K. Jauhar, H. Li, M. Sachan, S. Srivastava, and E.H. Hovy. 2013. A Structured Distributional Semantic Model for Event Co-reference. Poster, in *Proceedings of the Association of Computational Linguistics conference (ACL)*.

Goyal, K., S.K. Jauhar, H. Li, M. Sachan, S. Srivastava, and E.H. Hovy. 2013. A Structured Distributional Semantic Model: Integrating Structure with Semantics. Poster, in *Proceedings of the ACL Workshop on Continuous Vector Spaces and their Compositionality*, at the conference of the ACL.

Hovy, E.H., T. Mitamura, M.F. Verdejo, J. Araki, A. Philpot. 2013. Events are Not Simple: Identity, Non-Identity, and Quasi-Identity. *Proceedings of the 1st Events Workshop* at the conference of the HLT-NAACL.

Hsi, A., J.G. Carbonell, and Y. Yang. 2015. Modeling event extraction via multilingual data sources. *Proceedings of the TAC Evaluation Workshop*. Baltimore, MD.

Hsi, A., Jaime Carbonell, and Yiming Yang. 2016. CMU-CS event TAC-KBP 2016 event argument extraction system. *Proceedings of the Text Analysis Conference (TAC 2016).*

Hsi, A., Yiming Yang, Jaime Carbonell, and Ruochen Xu. 2016. Leveraging multilingual training for limited resource event extraction. *Proceedings of the COLING conference*.

Jauhar, S.K. and E.H. Hovy. 2014. Inducing Latent Semantic Relations for Structured Distributional Semantics. Oral paper, in *Proceedings of the COLING conference*.

Liu, Z.(H), Jun Araki, Eduard H. Hovy, and Teruko Mitamura. 2014. Supervised Within-Document Event Coreference using Information Propagation. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC).*

Liu, Z.(H), Jun Araki, Eduard Hovy, and Teruko Mitamura. 2014. Supervised Within-Document Event Coreference using Information Propagation. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC).*

Liu, Z.(H), Mitamura, T., and Hovy, E. 2015. Evaluation Algorithms for Event Nugget Detection : A Pilot Study. In *Proceedings of the 3rd Workshop on EVENTS at the NAACL-HLT*, pp. 53–57.

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

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

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

Ma, X., N.R. Fauceglia, Y. Lin, and E.H. Hovy. 2017. CMU system for entity discovery and linking at TAC-KBP 2017. In *Proceedings of Text Analysis Conference (TAC 2017)*.

Ma, X., Y. Gao, Z. Hu, Y. Yu, Y. Deng, and E.H. Hovy. 2017. Dropout with Expectation-linear Regularization. Poster, in *Proceedings of the ICLR conference*. Toulon, France.

Mazaitis, K., Richard C. Wang, Bhavana Dalvi, and William W. Cohen. 2014. A tale of two entity linking and discovery systems. *Proceedings of the Text Analysis Conference (TAC2014).*

Peng, H., Sam Thomson, and Noah A. Smith. 2017. Deep Multitask Learning for Semantic Dependency Parsing. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Schneider, N., Spencer Onuffer, Nora Kazour, Emily Danchik, Michael T. Mordowanec, Henrietta Conrad, and Noah A. Smith. 2014. Comprehensive Annotation of Multiword Expressions in a Social Web Corpus. *Proceedings of the 9th Linguistic Resources and Evaluation Conference*

Schneider, Nathan. 2014. *Lexical Semantic Analysis in Natural Language Text.* Ph.D. Thesis, September 2014.

Spiliopoulou, E., 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.

Srivastava, S., D. Hovy, and E.H. Hovy. 2013. A Walk-based Semantically Enriched Tree Kernel over Distributed Word Representations. *Proceedings of the EMNLP conference*.

Thomson, S., Brendan O'Connor, Jeffrey Flanigan, David Bamman, Jesse Dodge, Swabha Swayamdipta, Nathan Schneider, Chris Dyer and Noah A. Smith. 2014. CMU: Arc-Factored, Discriminative Semantic Dependency Parsing. *Proceedings of the SemEval Workshop*.

Srivastava, Shashank and Eduard H. Hovy. 2014. Vector Space Semantics with Frequency-driven Motifs. *Proceedings of the Association of Computational Linguistics conference (ACL) conference.*

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

Swayamdipta, S., Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-Semantic Parsing with Softmax-Margin Segmental RNNs and a Syntactic Scaffold. https://arxiv.org/pdf/1706.09528.

Xie, Q., X. Ma, Z. Dai, and E.H. Hovy. 2017. An Interpretable Knowledge Transfer Model for Knowledge Base Completion. *Proceedings of the ACL conference*. Vancouver, Canada.

Wang, William Yang, Kathryn Mazaitis, and William W. Cohen. 2014. Structure Learning via Parameter Learning. *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (CIKM 2014).*

Wang, William Yang, Kathryn Mazaitis, Ni Lao, and William W. Cohen. 2015. Efficient Inference and Learning in a Large Knowledge Base: Reasoning with Extracted Information using a Locally Groundable First-Order Probabilistic Logic. *Machine Learning Journal* (MLJ 2015), Springer.

Wang, William Yang, Lingpeng Kong, Kathryn Mazaitis, and William W. Cohen. 2014. Dependency Parsing for Weibo: An Efficient Probabilistic Logic Programming Approach. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014).*

Wang, William Yang, Kathryn Mazaitis, William W. Cohen. 2014. ProPPR: Efficient First-Order Probabilistic Logic Programming for Structure Discovery, Parameter Learning, and Scalable Inference. *Proceedings of the AAAI 2014 Workshop on Statistical Relational AI (StarAI 2014).*

Wang, W.Y. and W.W. Cohen. 2015. Joint Information Extraction and Reasoning: A Scalable

Statistical Relational Learning Approach. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and The 7th International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015),* long paper for oral presentation.

Wang, W.Y., K. Mazaitis, and W.W. Cohen. 2015. A Soft Version of Predicate Invention Based on Structured Sparsity. *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015),* full paper for oral presentation.

Wang, W.Y. and William W. Cohen. 2016. Learning first-order logic embeddings via matrix factorization. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (IJCAI16), Gerhard Brewka (Ed.). AAAI Press 2132-2138.

Wang, W.Y. and W.W. Cohen, 2016. Tutorial on Scalable Probabilistic Logics. To be presented at the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016).

Symbols, Abbreviations, and Acronyms

AMR: Abstract Meaning Representation
AIDA: Active Interpretation of Disparate Alternatives
Bi-LSTM: Bidirectional Long Short-Term Memory neural network design
CoNLL: Computational Natural Language Learning workshop/competition series
CoreNLP: Software package developed at Stanford University
CRF: Conditional Random Fields algorithm
DEFT: DARPA program name
EDL: Entity Detection and Linking task
GPE: GeoPolitical Entity formal representation name of semantic class
KB: Knowledge Base (database of semantic facts)
KBP: Knowledge Base Population challenge task
KRes: Knowledge Resolver algorithm used in PowerLoom
LSTM: Long Short-Term Memory neural network design
NER: Named Entity Recognition task
POS: Part of speech (tag)
PowerLoom: Knowledge representation and reasoning engine
SAFT: Semantic Analysis and Filtering of Text: project name
SEMAFOR: Semantic parser built by group in SAFT