AFRL-RI-RS-TR-2018-130

COMPOSING INFORMATION EXTRACTION, SEMANTIC PARSING AND TRACTABLE INFERENCE FOR DEEP NLP

UNIVERSITY OF WASHINGTON

MAY 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-2018-130 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
PETER ROCCI
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

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
MAY 2018	FINAL TECHNICAL REPORT	OCT 2012 – NOV 2017

4. TITLE AND SUBTITLE

COMPOSING INFORMATION EXTRACTION, SEMANTIC PARSING AND TRACTABLE INFERENCE FOR DEEP NLP

5a. CONTRACT NUMBER
FA8750-13-2-0019

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Daniel Weld, Pedro Domingos, Luke Zettlemoyer, Hannaneh Hajishirzi

5d. PROJECT NUMBER
DEFT

5e. TASK NUMBER
12

5f. WORK UNIT NUMBER
15

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Washington
Office of Sponsored Programs
4333 Brooklyn Ave NE
Seattle, WA 98195-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

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

Air Force Research Laboratory/RIED
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-2018-130

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

We developed new information extraction technologies. Our Vinculum entity linker is simple and modular; we compare it to other top systems analyze approaches to mention extraction, candidate generation, entity type prediction, entity co-reference, and coherence. We also developed both unsupervised and semi-supervised algorithms for event extraction that exploit parallel news streams, showing significant performance improvements on multiple event extractors over ACE 2005 and TAC-KBP 2015 datasets. Finally, we developed new natural language processing tools (e.g., semantic parsing) and introduced efficient inference algorithms for extracted knowledge bases

15. SUBJECT TERMS
Natural language processing, relation extraction, event extraction, knowledge discovery, probabilistic inference, semantic parsing

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Peter Rocci
U	U	U	UU	43	19b. TELEPHONE NUMBER (*Include area code*)

Contents

List of Figures

List of Tables

1 SUMMARY

This grant supported research on *information extraction* (IE) — the process of converting unstructured natural language text into semantically-meaningful structured data, such as might be stored in a relational database. We developed new methods for two types of IE: entity linking and event extraction, and open information extraction.

Entity Linking (EL) is a central (IE) task — given a textual passage, identify entity *mentions* (substrings corresponding to world entities) and link them to the corresponding entry in a given Knowledge Base. We developed a simple and modular, unsupervised EL system, VINCULUM, and compared it to the two leading sophisticated EL systems on a comprehensive set of nine datasets. While our system does not consistently outperform the best EL system, it does come remarkably close and serves as a competitive baseline for future research. Furthermore, we carry out an extensive ablation analysis, whose results illustrate 1) even a near-trivial model using CrossWikis [44] performs surprisingly well, and 2) incorporating a fine-grained set of entity types raises that level even higher.

Event extraction is the process of mapping a sentence describing an action or event to a relation between the entities involved, such as may be added to a relational database. Our work develops a new unsupervised technique, NEWSSPIKE-RE, to both discover event relations and extract them with high precision as well as a novel semi-supervised method. The intuition underlying NEWSSPIKE-RE is that the text of articles from two different news sources are not independent, since they are each conditioned on the same real-world events. By looking for rarely described entities that suddenly "spike" in popularity on a given date, one can identify paraphrases. Such *temporal correspondence* [49] allow one to cluster diverse sentences, and the resulting clusters may be used to form training data in order to learn event extractors. Furthermore, one can also exploit parallel news to obtain direct *negative* evidence. Our NEWSSPIKE-RE system encapuslates these intuitions in a novel graphical model. Experiments demonstrate that it has extremely high performance. Furthermore, we introduce and evaluate a semi-supervised method for combining labeled and unlabeled data for event extraction, showing significant performance improvements on multiple event extractors over ACE 2005 and TAC-KBP 2015 datasets.

In addition we made fundamental contributions to numerous NLP tools used in these high-level tasks, including better coreference detection, semantic parsing and tractable inference.

2 INTRODUCTION

This grant supported research on *information extraction* (IE) — the process of converting unstructured natural language text into semantically-meaningful structured data, such as might be stored in a relational database. We developed new methods for two types of IE: entity linking and event extraction, and open information extraction. The simplest form of IE is *entity linking* — mapping a textual substring into the corresponding entity in a background knowledge base. *Event extraction* is the process of mapping a sentence describing an action or event to a relation between the entities involved, such as may be added to a relational database. In addition to our core work on IE, we also developed new natural language processing tools (semantic parsing) and worked on efficient algorithms for inference over extracted knowledge bases.

2.1 Entity Linking

Entity Linking (EL) is a central Information Extraction (IE) task — given a textual passage, identify entity *mentions* (substrings corresponding to world entities) and link them to the corresponding entry in a given Knowledge Base (KB, e.g. Wikipedia or Freebase). For example,

JetBlue begins direct service between Barnstable Airport and JFK International.

Here, "JetBlue" should be linked to the entity `KB:JetBlue`, "Barnstable Airport" to `KB:Barnstable Municipal Airport`, and "JFK International" to `KB:John F. Kennedy International Airport`[1]. The links not only provide semantic annotations to human readers but also a machine-consumable representation of most basic semantic knowledge in the text. Many other NLP applications can benefit from such links, such as distantly-supervised relation extraction [5, 41, 21, 24] that uses EL to create training data, and some coreference systems that use EL for disambiguation [15, 52]. Unfortunately, in spite of numerous papers on the topic and several published data sets, there is surprisingly little understanding about state-of-the-art performance.

We argue that there are three reasons for this confusion. First, *there is no standard definition of this problem.* A few variants have been studied in the literature, such as Wikification [36, 40, 3] which aims at linking noun phrases to Wikipedia entities and Named Entity Linking (aka Named Entity Disambiguation) [32, 20] which targets only named entities. Here we use the term *Entity Linking* as a unified name for both problems, and *Named Entity Linking* (NEL) for the subproblem of linking only named entities. But names are just one part of the problem. For many variants there are no annotation guidelines for scoring links. What types of entities are valid targets? When multiple entities are plausible for annotating a mention, which one should be chosen? Are nested mentions allowed? Without agreement on these issues, a fair comparison is elusive.

Secondly, *it is almost impossible to assess approaches, because systems are rarely compared using the same data sets.* For instance, Hoffart *et al.* [20] developed a new data set (AIDA) based on the CoNLL 2003 Named Entity Recognition data set but failed to evaluate their system on MSNBC previously created by [6]; Wikifier [3] compared to the authors' previous system [40] using the originally selected datasets but didn't evaluate using AIDA data.

Finally, when two end-to-end systems are compared, *it is rarely clear which aspect of a system makes one better than the other.* This is especially problematic when authors introduce complex mechanisms or nondeterministic methods that involve learning-based reranking or joint inference. To address these problems, we analyzed several significant inconsistencies among the data sets and suggest resolutions, which we hope can facilitate construction of a consistent annotation guide- line in the near future. To have a better understanding of the importance of various techniques, we developed a simple and modular, unsupervised EL system, VINCULUM. We compared VINCULUM to the two leading sophisticated EL systems on a comprehensive set of nine datasets. While our system does not consistently outperform the best EL system, it does come remarkably close and serves as a competitive baseline for future research. Furthermore, we carry out an extensive ablation analysis, whose results illustrate 1) even a near-trivial model using CrossWikis [44] performs surprisingly well, and 2) incorporating a fine-grained set of entity types raises that level even higher. In summary, our work on entity linking makes the following contributions:

[1] We use typewriter font, e.g., `KB:Entity`, to indicate an entity in a particular KB, and quotes, e.g., "Mention", to denote textual mentions.

- We analyze and discuss the differences among several versions of the entity linking problem, compare existing data sets, and suggest guidelines for future data annotation.
- We present a simple yet effective, modular, unsupervised system, VINCULUM, for entity linking. We make the implementation open source and publicly available for future research.
- We compare VINCULUM to 2 state-of-the-art systems on an extensive evaluation of 9 data sets. We also investigate several key aspects of the system including mention extraction, candidate generation, entity type prediction, entity coreference, and coherence between entities.

2.2 Relation & Event Extraction

Relation extraction, the process of extracting structured information from natural language text, grows increasingly important for Web search and question answering. Event extraction is similar, but the relations correspond to events. Traditional supervised approaches, which can achieve high precision and recall, are limited by the cost of labeling training data and are unlikely to scale to the thousands of relations on the Web. Another approach, distant supervision [5, 46], creates its own training data by matching the ground instances of a Knowledge base (KB) (*e.g.* Freebase) to the unlabeled text.

Unfortunately, while distant supervision can work well in some situations, the method is limited to relatively *static* facts (*e.g., born-in(person, location)* or *capital-of(location,location)*) where there is a corresponding knowledge base. But what about dynamic *event relations* (also known as *fluents*), such as *travel-to(person, location)* or *fire(organization, person)*? Since these time-dependent facts are ephemeral, they are rarely stored in a pre-existing KB. At the same time, knowledge of real-time events is crucial for making informed decisions in fields like finance and politics. Indeed, news stories report events almost exclusively, so learning to extract events is an important open problem.

Our work develops a new unsupervised technique, NEWSSPIKE-RE, to both discover event relations and extract them with high precision as well as a novel semi-supervised method. The intuition underlying NEWSSPIKE-RE is that the text of articles from two different news sources are not independent, since they are each conditioned on the same real-world events. By looking for rarely described entities that suddenly "spike" in popularity on a given date, one can identify paraphrases. Such *temporal correspondence* [49] allow one to cluster diverse sentences, and the resulting clusters may be used to form training data in order to learn event extractors. Furthermore, one can also exploit parallel news to obtain direct *negative* evidence. To see this, suppose one day the news includes the following: (a) *"Snowden travels to Hong Kong, off southeastern China."* (b) "*Snowden cannot stay in Hong Kong as Chinese officials will not allow ..."* Since news stories are usually coherent, it is highly unlikely that *travel to* and *stay in* (which is negated) are synonymous. By leveraging such direct negative phrases, we can learn extractors capable of distinguishing heavily co-occurring but semantically different phrases, thereby avoiding many extraction errors. Our NEWSSPIKE-RE system encapuslates these intuitions in a novel graphical model making the following contributions:

- We develop a method to discover a set of distinct, salient event relations from news streams.
- We describe an algorithm to exploit parallel news streams to cluster sentences that belong to the same event relations. In particular, we propose the *temporal negation heuristic* to avoid conflating co-occurring but non-synonymous phrases.

- We introduce a probabilistic graphical model to generate training for a sentential event extractor without requiring any human annotations.
- We introduce a semi-supervised method for combining labeled and unlabeled data for event extraction, showing significant performance improvements on multiple event extractors over ACE 2005 and TAC-KBP 2015 datasets.

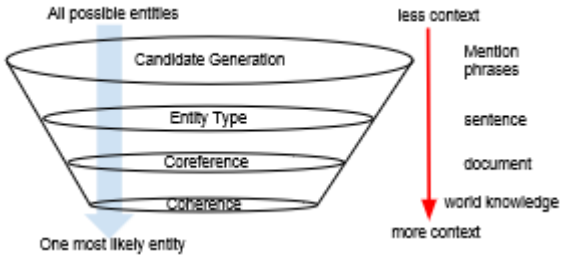

Figure 1: The process of finding the best entity for a mention. All possible entities are sifted through as VINCULUM proceeds at each stage with a widening range of context in consideration.

2.3 Supporting Tools and Methods

In addition we made fundamental contributions to numerous NLP tools used in these high-level tasks, including better coreference detection, semantic parsing and tractable inference.

3 METHODS, ASSUMPTIONS, AND PROCEDURES

3.1 Entity Linking

In this section, we present VINCULUM, a simple, unsupervised EL system that performs comparably to the state of the art. As input, VINCULUM takes a plain-text document d and outputs a set of segmented mentions with their associated entities $A_d = \{(m_i, l_i)\}$. VINCULUM begins with mention extraction. For each identified mention m, candidate entities $C_m = \{c_j\}$ are generated for linking. VINCULUM assigns each candidate a linking score $s(c_j | m, d)$ based on the entity type compatibility, its coreference mentions, and other entity links around this mention. The candidate entity with the maximum score, *i.e.* $l = \arg\max_{c \in C_m} s(c | m, d)$, is picked as the predicted link of m.

Figure 1 illustrates the linking pipeline that follows mention extraction. For each mention, VINCULUM ranks the candidates at each stage based on an ever widening context. For example, candidate generation (Section 3.1) merely uses the mention string, entity typing (Section 3.1) uses the sentence, while coreference (Section 3.1) and coherence (Section 3.1) use the full document and Web respectively. Our pipeline mimics the sieve structure introduced in [26], but instead of merging coreference clusters, we adjust the probability of candidate entities at each stage. The modularity of VINCULUM enables us to study the relative impact of its subcomponents.

Mention Extraction The first step of EL extracts potential mentions from the document. Since VINCULUM restricts attention to named entities, we use a Named Entity Recognition (NER)

system [12]. Alternatively, an NP chunker may be used to identify the mentions.

Dictionary-based Candidate Generation

While in theory a mention could link to any entity in the KB, in practice one sacrifices little by restricting attention to a subset (dozens) precompiled using a dictionary. A common way to build such a dictionary D is by crawling Web pages and aggregating anchor links that point to Wikipedia pages. We adopt the CrossWikis dictionary, which was computed from a Google crawl of the Web [44]; the frequency with which a mention, m, links to a particular entity, c, allows one to estimate the conditional probability $p(c|m)$.

In addition, we employ two small but precise dictionaries for U.S. state abbreviations and demonyms when the mention satisfies certain conditions. For U.S. state abbreviations, a comma before the mention is required. For demonyms, we ensure that the mention is either an adjective or a plural noun.

Incorporating Entity Types For an ambiguous mention such as "Washington", knowing that the mention denotes a person allows an EL system to promote `KB:George Washington` while lowering the rank of the capital city in the candidate list. We incorporate this intuition by combining it probabilistically with the CrossWikis prior.

$$p(c|m,s) = \sum_{t \in T} p(c,t|m,s) = \sum_{t \in T} p(c|m,t,s)p(t|m,s) \ ,$$

where S denotes the sentence containing this mention m and T represents the set of all possible types. We assume the candidate c and the sentential context s are conditionally independent if both the mention m and its type t are given. In other words, $p(c|m,t,s) = p(c|m,t)$, the RHS of which can be estimated by renormalizing $p(c|m)$ w.r.t. type t:

$$p(c|m,t) = \frac{p(c|m)}{\sum_{c \to t} p(c|m)} \ ,$$

where $c \to t$ indicates that t is one of c's entity types.[2] The other part of the equation, $p(t|m,s)$, can be estimated by any off-the-shelf Named Entity Recognition system, *e.g.* [12] and [29].

Coreference It is common for entities to be mentioned more than once in a document. Since some mentions are less ambiguous than others, it makes sense to use the most representative mention for linking. To this end, VINCULUM applies a coreference resolution system (*e.g.* [26]) to cluster mentions that are coreferent. The representative mention of such a cluster is chosen for linking.[3] While there are more sophisticated ways to integrate EL and coreference [15], VINCULUM's pipeline is simple and modular.

[2] We notice that an entity often has multiple appropriate types, e.g. a school can be either an organization or a location depending on the context. We use Freebase to provide the entity types and map them appropriately to the target type set.

[3] When two mentions overlap, we choose the one without a relative clause, which is favorable for candidate generation.

Coherence When KB:Barack Obama appears in a document, it is more likely that the mention "Washington" represents the capital KB:Washington, D.C. as the two entities are semantically related, and hence the joint assignment is *coherent*. A number of researchers found inclusion of some version of coherence is beneficial for EL [6, 36, 40, 20, 3]. For incorporating it in VINCULUM, we seek a document-wise assignment of entity links that maximizes the sum of the coherence scores between each pair of entity links $coh(c_{m_i}, c_{m_j})$ predicted in the document d, *i.e.* $\sum_{m_i, m_j \in M_d, i \neq j} \phi(l_{m_i}, l_{m_j})$ where φ is a function that measures the coherence between two entities and l_{m_i} (l_{m_j}) is one of the candidates of m_i(m_j). Instead of searching for the exact solution in a brute-force manner ($O(|C|^{|M|})$ where $|C| = \max_m |C_m|$), we isolate each mention and greedily look for the best candidate by fixing the predictions of other mentions, allowing linear time search $(O(|C| \cdot |M|))$.

Specifically, for a mention *m* and each of its candidates, we compute a score

$$coh(c) = \frac{1}{|P_d|-1} \sum_{p \in P_d \setminus \{p_m\}} \phi(p, c), c \in C_m,$$

where P_d is the union of all intermediate links $\{p_m\}$ in the document.

Since both measures take values between 0 and 1, we denote the coherence score $coh(c)$ as $p_\varphi(c|P_d)$, the conditional probability of an entity given other entities in the document. The final score of a candidate is the sum of coherence $p_\varphi(c|P_d)$, and type compatibility $p(c|m, s)$.

Two coherence measures have been found to be useful: Normalized Google Distance (NGD) [36,40] and relational score [3]. NGD between two entities c_i and c_j is defined based on the link structure between Wikipedia articles as follows:

$$\phi_{NGD}(c_i, c_j) = 1 - \frac{\log(\max(|L_i|,|L_i|)) - \log(|L_i \cap L_j|)}{\log(W) - \log(\min(|L_i|,|L_i|))}$$

where L_i and L_j are the incoming (or outgoing) links in the Wikipedia article for c_i and c_j respectively and W is the total number of entities in Wikipedia. The relational score between the two entities is a binary indicator whether a relation exists between them. We use Freebase[4] as the source of the relation triples *F={(sub, rel, obj)}*. Relational coherence is defined as:

$$\phi_{REL}(e_i, e_j) = \begin{cases} 1 & \exists r, (e_i, r, e_j) \text{ or } (e_j, r, e_i) \in F \\ 0 & \text{otherwise.} \end{cases}$$

[4]The mapping between Freebase and Wikipedia is provided at https://developers.google.com/freebase.

3.2 Relation & Event Extraction

The UW event extraction system is composed of three separate systems which can each operate as independent event and argument extractors. Two of these systems were newly developed for the 2016 TAC KBP; the third, NewsSpike, is an existing UW event extractor.

Evento is a CRF-based structured event and argument extractor. It takes a pipelined approach in which each stage of the pipeline uses a loss-augmented training function allowing it to be tuned to improve either precision or recall.

NomEvent is a supervised extractor with a focus on extracting events triggered by nouns. It uses a lexicon of likely nominal event triggers generated through an automated process (described below) to generate features.

NewsSpike is trained using an unsupervised process based upon OpenIE principles, so the set of events it extracts is not based upon the Rich ERE (RERE) ontology [48]. In order to participate in the TAC KBP evaluation, a mapping was created from NewsSpike events to RERE. Only a subset of NewsSpike's events could be mapped to RERE events, so NewsSpike served as a low recall but high precision contributor to the overall system.

Both Evento and NomEvent use a pipelined approach, in which a document is passed through the following process: (1) Preprocessing with Stanford CoreNLP (POS, NER, dependency parsing, and lemmatization), (2) Entity Extraction, (3) Trigger Extraction and Classification, (4) Argument Classification, (5) Realis Classification.

The preprocessing step is identical for both systems, but they differ in the remaining steps. Both systems use linear classifiers to perform trigger and argument classification, but differ in the features used in their classifiers, as shown in Table 1. Evento and NomEvent were both trained on the ACE 2005 corpus, while NomEvent's training data was also supplemented with Rich ERE data from LDC2016E60.

Table 1: A comparison of features used in the trigger classifiers of Evento and NomEvent

Trigger Features	
Evento & NomEvent	Token bigram Dependency bigram Dependent lemma Governor lemma NER types in sentence Entity types in sentence POS tags
Evento only	Basic WordNet Synonyms Brown Clusters
NomEvent only	Token Word2Vec embedding Dependency path to sentence nouns Document-level Event Basket hits Event Basket Bag of Words Event Basket Distance Comparison WordNet lexname WordNet traversal features

3.2.1 Evento

Evento is a supervised system that uses a structured model with features primarily based on those used by [27], which is the current state-of-the-art for models with discrete features.

3.2.2 Entity Extraction

Entity extraction in Evento uses a semi-Markov conditional random field. Given a sentence $x = (x_1, \ldots, x_n)$ the model considers sequences of labeled spans $\bar{s} = ((\boldsymbol{l}_1, b_1, e_1), (\boldsymbol{l}_2, b_2, e_2), \ldots, (\boldsymbol{l}_k, b_k, e_k))$, where $\boldsymbol{l}_i \in \{\text{Entity}, \text{Non-Entity}\}$ is a label for each span and $b_i, e_i \in \{0, 1 \ldots n\}$ are fenceposts for

Table 2: Features used during the entity extraction step of Evento. Word properties capture information about the capitalization, numbers, and punctuation in a word.

Evento Entity Features	
Word Features	Span Features
Word properties	Token n-gram context
Token	Span length
Prefixes	Dependency arcs entering/leaving span
Suffixes	Phrase type of span in constituency tree

each span such that $b_i < e_i$ and $e_i = b_{i+1}$.

The model places distributions over these sequences given the sentence as follows:

$$p_\theta(\bar{s}|x) \propto \exp\left(\theta^\top \sum_{i=1}^{k} f(x, (\ell_i, b_i, e_i))\right)$$

where *f* is a feature function that computes features for a span given the input sentence. The feature function we use includes both the union of token level features fired for each token in a span as well as features fired for the overall span. The specific features we use are outlined in Table 2.

We train this model on the gold entity annotations found in the ACE 2005 corpus[2]. In order to train the model, we maximize the conditional log likelihood of the training data augmented with a loss function via softmax-margin. We optimize using the AdaGrad algorithm with L_2 regularization.

Evento Trigger Extractor Both the trigger and argument classification stages in Evento are performed using linear-chain conditional random fields (CRF). Similar to entity extraction, we train the models by maximizing the conditional log likelihood of the training data augmented with a loss function and optimize using AdaGrad with L_2 regularization. During trigger extraction, each token in a sentence is assigned a label. Each label is either an event type we are interested in or *NO-EVENT* signifying that the token is not a trigger. The features we use for trigger classification are given in Table 1.

Evento Argument Extractor As mentioned in the previous section, we use a CRF to perform argument classification. For every trigger identified in the previous step, the system assigns argument roles to each entity in the sentence. The possible roles depend on what arguments a particular event can take, as well as *NO-ARGUMENT* signifying that the entity did not participate in the event. Note that multiple triggers can occur in a sentence, so the system may have to classify an entity multiple times for separate event triggers. The features we used are outlined in Table 3.

3.2.3 NomEvent

The motivation behind NomEvent is to use existing NLP resources to develop an event extraction system focused on identifying events triggered by nouns.

We first aim to develop a lexicon of likely nominal event triggers by starting with a seed verb corresponding to an event and then searching WordNet and FrameNet for related nominal forms.

[2] https://www.ldc.upenn.edu/collaborations/past-projects/ace

Table 3: Features used during the argument extraction step of Evento.

Evento Argument Features
Token bigrams
POS bigrams
Distance to trigger word
Dependency path to trigger word
NER Tags

Table 4: The seed verb and synsets used on the creation of the NomEvent lexicon entry for Conflict.Attack are shown, along with some of the twenty-six words in the resulting lexicon.

NomEvent Lexicon Generation	
Event	Conflict.Attack
Seed verb	attack
Hand-selected synsets	attack.n.01 attack.v.01 attack.v.06
Resulting Lexicon (selections)	Occupation Offensive Onrush Raid Storm Strike Torpedo

This lexicon is then used to build features for a supervised classifier. A pre-trained Google Word2Vec model trained on Google News data was used in developing the lexicon and in the classifier.

NomEvent was developed with a focus on detecting events triggered by nouns, but the process was adapted detect events triggered by verbs as well. Both a nominal-trigger-only NomEvent system and an all-trigger NomEvent system were used in the evaluation, as described below in the descriptions of each run.

3.2.4 Lexicon Construction

In order to develop the lexicon of potential nominal event triggers, we start with a seed verb for each event in the ontology. In most cases, the event subtype is used as the seed verb, such as “attack” for Conflict.Attack or “meet” for Contact.Meet. In cases where the event subtype was not suitable as a single verb, such as for Personnel.Start-Position, a human user selected a word to use as a seed (in this example, “hire”). For a few events, there was not a single obvious word that completely

characterized the event, such as Personnel.EndPosition; potential verbs for this event might include "quit", "fire", or "layoff". For the system presented here, one of these words was selected by the user (in this example, "quit").

Once a seed verb has been selected for each event, a human user searched WordNet for synsets in which that word participated and selected one or more synsets with definitions that best characterized that event. This operation took less than 5 minutes of user time per event. An alternate version was explored where the most common synset for each seed verb was used, but this method was found to produce less accurate results.

Once a selection of synsets is made, all lemmas for each synset were retrieved, along with immediate hyponyms. For each lemma on the resulting list, the cosine distance between the Word2Vec embeddings for that lemma and its corresponding seed verb was calculated, and any lemma with a distance greater than 0.75 was discarded. For each remaining lemma, all derivationally-related nouns are then added to the lexicon. (For the version of the system which identifies triggers from any part of speech, all related forms are added). Table 4 shows the seed verb and synsets used for the Conflict.Attack event, as well as selected nouns in the resulting lexicon.

When a lemma is added to the lexicon, an additional set of features related to that lemma are also saved for use in the trigger classifier. These features, listed on Table 1 as "WordNet traversal features", include the cosine distance to the seed verb, the total number of times that lemma appeared in the WordNet traversal for that seed verb, the WordNet corpus frequency for that lemma, and the percent of all corpus mentions for that synset which that lemma represented.

In addition, a FrameNet search is made for each seed verb. If a frame is found which matches the event, any nouns participating in that frame are added to the lexicon if not already present.

Entity Extraction A CRF-based entity extractor was trained on the ACE 2005 corpus. Features included part-of-speech, NER tag, and word shape.

NomEvent Trigger Extractor The NomEvent trigger extractor uses an L2-regularized multil-abel logistic regression classifier to process each word in a sentence in sequence and apply a label (either an event subtype or "None"). In addition to conventional features (summarized on Table 1), several features are calculated based on the lexicon. We use "Event Basket" to designate the words in the lexicon associated with each event.

The first lexicon-based feature is the number of words in the document which are present in each Event Basket. This feature is based on the observation that many newswire documents constitute a narrative which repeatedly references elements of an event throughout the article, so potential triggers which appear in a surrounding context containing many words related to that event are highly likely to relate to an event.

The second lexicon-based feature is a simple binary vector indicating if the token lemma matched any of the lexicon words. If one of the words was matched, the features described above that were gathered in the traversal of WordNet are included as well.

The final lexicon-based feature is, for each event, the average cosine distance between the word embedding for the token lemma and the embedding for each word in the event basket for that event. This provides a score for each event which represents the distance of the token to the set of embeddings which represent that event.

NomEvent Argument Extractor When a trigger has been identified in a sentence, all entities in that sentence are classified to determine whether they are argument for the event. The argument classifier is identical to the trigger classifier, but with the addition of several features: Dependency

path to trigger, dependency path length, distance to trigger, entity type, NER type of previous and next word, and whether the prior or next lemma were on a small hand-collected list of words such as “the”, “to”, and “of”. In addition, the event basket Word2Vec similarity comparison was replaced with Word2Vec comparisons with a small manually generated list of words related to event roles such as “city”, “company”, and “attacker”.

3.2.5 Adapting NewsSpike to TAC Ontology

NewsSpike is an event extractor which uses an unsupervised method based on Open Information Extraction principles to generate and cluster data on which it is trained [48]. The set of events on which it is trained is thus discovered from data and does not correspond to a pre-set ontology. In order to adapt NewsSpike to the Rich ERE ontology, a mapping was created to map each of the 150 events which NewsSpike extracts to an RERE event, or to null if no RERE event existed. This mapping was completed manually. Of the 150 NewsSpike event, only 65 could be mapped to the events in the ontology for this year’s EAL and Nugget evaluations. Many NewsSpike events are fine-grained, so of these 65, many only partially corresponded to the ERE counterpart; for example, the NewsSpike events “apologize”, “assure”, “congratulate”, “reach out”, “talk”, and “warn” were all mapped to Contact.Correspondence. The existing NewsSpike system was trained on a wide ranging corpus of news articles scraped from the web. A version of NewsSpike trained on a domain better corresponding to the TAC ontology would likely provide more meaningful results.

3.3 Semi-Supervised Event Extraction

We also developed a method for self-training event extraction systems by bootstrapping additional training data. This is done by taking advantage of the occurrence of multiple mentions of the same event instances across newswire articles from multiple sources in much the same way as performed in our unsupervised approach. If our system can make a high-confidence extraction of some mentions in such a cluster, it can then acquire diverse training examples by adding the other mentions as well. Our goal is to automatically add high quality labeled examples, which can then be used as additional training data to improve the performance of any event extraction model. Our data generation process has three steps. The first is to identify clusters of news articles all describing the same event. The second step is to run a baseline system over the sentences in these clusters to identify events found in each cluster. Finally, once we have identified an event in one article in a cluster, our system scans through the other articles in that cluster choosing the most likely trigger in each article for the given event type.

Cluster Articles In order to identify groups of articles describing the same event instance, we use an approach inspired by the NewsSpike idea introduced in [48]. The main intuition is that rare entities that are mentioned a lot on a single date are more indicative that two articles are covering the same event. We assign a score, S, to each pair of articles, (a_i, a_j) appearing on the same day, for whether or not they cover the same event, as follows:

$$S(a_i, a_j) = \sum_{e \in E_{a_i} \cap E_{a_j}} \frac{\text{count}(e, \text{date}_{a_i,a_j})}{\text{count}(e, \text{corpus})},$$

where E_a is the list of named entities for the article a, and count is the number of times the entity appears on the given date, or in the whole corpus. This follows from the intuition above by reducing the weight given to common entities. For example, *United States* appears 367k times in the corpus, so it is not uncommon for it to appear hundreds of times on a single day, and articles mentioning it could be covering completely different topics. Meanwhile *Les Miles* appears only 1.6k times in the corpus, so when there are hundreds of mentions involving *Les Miles* on a single day, it is much more likely that he participated in some event. Accumulating these counts over all shared entities between two articles thus indicates whether the articles are covering the same event. We then group all articles that cover the same event according to this score into clusters.

Label Clusters Then, given clusters of articles, we run a baseline extractor which was trained on what limited amount of fully-supervised training data is available. The hope is that one or more of a cluster's sentences will use language similar enough to our training data that the extractor can make an accurate prediction. Our system keeps any cluster in which the baseline system identifies at least some threshold, θ_{event}, of event mentions for a single event type, and labels those clusters with the identified type.

Assign Triggers After labeling, the event clusters are comprised of articles in which at least one sentence should contain event mentions of the labeled type. Because most current event extraction systems require labeled event triggers for sentences, we identify those sentences and the event triggers therein so that we can run the baseline systems. For each sentence we identify the most likely trigger by checking the similarity of the word embeddings to the canonical vector for that event. This vector is computed as the average of the embeddingsof the event triggers, vt, in the gold training data:

$$v_{event} = \frac{1}{|T_{event}|} \sum_{t \in T_{event}} v_t,$$

where T_{event} is the set of triggers for this event in the gold training data. If the maximum similarity is greater than some threshold, θ_{sim}, the sentence and the corresponding trigger are added to the training data.

Event Trigger Identification Systems Event extraction tasks such as ACE and TAC-KBP have frequently been approached with supervised machine learning systems based on hand-crafted features, such as the system adapted from [27] which we make use of here. Recently, state-of-the-art results have been obtained with neural-network-based systems [37, 2, 9]. Here, we make use of two systems whose implementations are publicly available and show that adding additional data would improve their performance.

The first system is the joint recurrent neural net (JRNN) introduced by [37]. This model uses a bi-directional GRU layer to encode the input sentence. It then concatenates that with the vectors of words in a window around the current word, and passes the concatenated vectors into a feed-forward network to predict trigger types for each token. Because we are only classifying triggers, and not arguments, we don't include the memory vectors/matrices, which primarily help improve argument prediction, or the argument role prediction steps of that model.

The second is a conditional random field (CRF) model with the trigger features introduced by [27]. These include lexical features, such as tokens, part-of-speech tags, and lemmas, syntactic features, such as dependency types and arcs associated with each token, and entity features, including unigrams/bigrams normalized by entity types, and the nearest entity in the sentence. In particular, we use the Evento system from [10].

3.4 Coreference Resolution

Many errors in coreference resolution come from semantic mismatches due to inadequate world knowledge. Errors in named-entity linking (NEL), on the other hand, are often caused by superficial modeling of entity context. We demonstrate that these two tasks are complementary. We introduce a new model for named entity linking and coreference resolution, which solves both problems jointly, reducing the errors made on each. NECO extends the Stanford deterministic coreference system by automatically linking mentions to Wikipedia and introducing new NEL-informed mention-merging sieves. Linking improves mention-detection and enables new semantic attributes to be incorporated from Freebase, while coreference provides better context modeling by propagating named-entity links within mention clusters. We show consistent improvements across a number of datasets and experimental conditions, including over 11% reduction in MUC coreference error and nearly 21% reduction in F1 NEL error on ACE 2004 newswire data.

3.5 CCG Parsing to Build Semantic Structures

We have developed and publicly release a large variety of different core semantics models to drive the downstream work on high quality knowledge base population. Our central avenue of work has focus on building CCG parsers that can be used to do a wide variety of different styles of semantic analysis. This includes approaches for using these parsers to detect and resolve time expressions (Lee et al 2014), do information extraction (Choi, et al 2015), build SRL structures (Lewis et al, 2015), and build AMR graphs (Artzi et al, 2015). The algorithms that were developed improved as we progressed through the work period leading to two key new parsing algorithms that I will describe in more detail below. These algorithms remain some of the highest accuracy and fastest semantic parsers to date.

Given the advances in neural methods in other areas around 2015, it was natural to ask how to build the best neural CCG parser. Here, we build on recent supper tagging successes and build a parser that relied almost exclusively on an LSTM tagging model. We demonstrated that a state-of-the-art parser can be built using only a lexical tagging model and a deterministic CCG grammar, with no explicit model of bi-lexical dependencies (Lewis et al, 2016). Instead, all dependencies are implicitly encoded in an LSTM super-tagger that assigns CCG lexical categories. The parser significantly outperforms all previously published CCG results, supports efficient and optimal A decoding, and benefits substantially from semi-supervised tri-training. We did a detailed analysis, demonstrating that the parser can recover long range dependencies with high accuracy and that the semi-supervised learning enables significant accuracy gains. By running the LSTM on a GPU, we were able to parse over 2600 sentences per second while improving state-of-the-art accuracy by 1.1 F1 in domain and up to 4.5 F1 out of domain.

Following up on this result, we also investigated how to build more global factors back into the model, to see if we could further improve accuracy. We introduced the first global recursive neural parsing model with optimality guarantees during decoding (Lee et al, 2016). To support global features, we gave up dynamic programs and instead searched directly in the space of all possible subtrees. Although this space is exponentially large in the sentence length, we showed it is possible to learn an efficient A* parser. We augmented existing parsing models, which have informative bounds on the outside score, with a global model that has loose bounds but only needs to

modelnon-local phenomena. The global model is trained with a new objective that encourages the parser to explore a tiny fraction of the search space. This approach was used to directly expand our parser described above, further improving state-of-the-art accuracy by 0.4 F1. The final parser finds the optimal parse for 99.9

Together, this line of work provided very strong tools for work in knowledge base completion. We also trained CCG parsers for Chinese, which provided strong accuracy. However, as work continued it became more clear that neural methods were even more likely to displace work in syntactic parsing with CCG. In the last year of the program, we shifted to developing end-to-end models for classic problems in broad coverage semantics.

3.6 End-to-end Deep Learning

Our deep learning approaches focus on semantic role labeling and coreference resolution. In each case, we were able to show that simple methods can directly predict the desired semantic structures, without using the typically NLP pipeline (e.g. POS tagging, parsing, etc.). These models were trained on the OntoNotes corpus, and we also verified that they work well on Chinese and Arabic, setting new state of the art performance levels in both cases (there results are not in the published papers, because they were done only very recently).

We first studied the problem of PropBank semantic role labeling. We introduced a new deep learning model that significantly improves the state of the art, along with detailed analyses to reveal its strengths and limitations (He et al, 2017). We used a deep highway BiLSTM architecture with constrained decoding, while observing a number of recent best practices for initialization and regularization. Our 8-layer ensemble model achieves 83.2 F1 on the CoNLL 2005 test set and 83.4 F1 on CoNLL 2012, roughly a 10

Building on this result, we also investigated whether we could perform a similar feat for coreference resolution. We introduce the first end-to-end coreference resolution model and show that it significantly outperforms all previous work without using a syntactic parser or hand- engineered mention detector (Lee et al, 2017). The key idea is to directly consider all spans in a document as potential mentions and learn distributions over possible antecedents for each. The model computes span embeddings that combine context-dependent boundary representations with a head-finding attention mechanism. It is trained to maximize the marginal likelihood of gold antecedent spans from coreference clusters and is factored to enable aggressive pruning of potential mentions. Experiments demonstrate state-of-the-art performance, with a gain of 1.5 F1 on the OntoNotes benchmark and by 3.1 F1 using a 5-model ensemble, despite the fact that this is the first approach to be successfully trained with no external resources.

Although this work was primarily in the last reporting period, it sets the tone for how we expect modeling would proceed future work for all of the related problems we developed for our full DEFT research efforts, as long as there is sufficient labeled training data.

3.7 Semantic Training Resources

In addition to developing new models, we have also focused on gather new resources for training high capacity neural models at large scale, and for incorporating external resources into already trained models. Here, we have looked at using external knowledge bases to provide better supervision

for relation extraction models (Ritter et al, 2013) and to provide background knowledge for joint models of coreference resolution and entity linking (Hajishirzi et al, 2013). We also introduced new datasets for non-propositional language understanding tasks, including factuality predication (Lee et al, 2015) and document-level models of entity-entity sentiment (Choi et al, 2016). However, I will highlight the data efforts that provide rich semantic structures that can be used to do detailed semantic analysis at the sentence and paragraph level, using different forms of questions answer pairs as task supervision.

Our first effort focused on using question answering as a proxy for semantic role labeling. We introduced the task of question-answer driven semantic role labeling (QA-SRL), where question-answer pairs are used to represent predicate-argument structure (He et al, 2015). For example, the verb ”introduce” in the previous sentence would be labeled with the questions ”What is introduced?”, and ”What introduces something?”, each paired with the phrase from the sentence that gives the correct answer. Posing the problem this way allows the questions themselves to define the set of possible roles, without the need for predefined frame or thematic role ontologies. It also allows for scalable data collection by annotators with very little training and no linguistic expertise. We gathered data in two domains, newswire text and Wikipedia articles, and introduced simple classifier-based models for predicting which questions to ask and what their answers should be. Our results showed that non-expert annotators can produce high quality QA-SRL data, and also establish baseline performance levels for future work on this task.

We also looked at the more general case of questions answering for reading comprehension. We developed TriviaQA, a challenging reading comprehension dataset containing over 650K question answer-evidence triples (Joshi et al, 2017). TriviaQA includes 95K question-answer pairs authored by trivia enthusiasts and independently gathered evidence documents, six per question on average, that provide high quality distant supervision for answering the questions. We showed that, in comparison to other recently introduced large-scale datasets, TriviaQA (1) has relatively complex, compositional questions, (2) has considerable syntactic and lexical variability between questions and corresponding answer-evidence sentences, and (3) requires more cross sentence reasoning to find answers. We also presented two baseline algorithms: a feature-based classifier and a state-of-the-art neural network, that performs well on SQuAD reading comprehension. Neither approach comes close to human performance (23% and 40

Finally, we combined these ideas by showing that relation extraction can be reduced to answering simple reading comprehension questions, by associating one or more natural-language questions with each relation slot (Levy et al 2017). This reduction has several advantages: we can (1) learn relation-extraction models by extending recent neural reading-comprehension techniques, (2) build very large training sets for those models by combining relation-specific crowd-sourced questions with distant supervision, and even (3) do zero-shot learning by extracting new relation types that are only specified at test-time, for which we have no labeled training examples. Experiments on a Wikipedia slot-filling task demonstrated that the approach can generalize to new questions for known relation types with high accuracy, and that zero-shot generalization to unseen relation types is possible, at lower accuracy levels, setting the bar for future work on this task.

3.8 Tractable Markov Logic (TML)

We have developed a framework for tractable probabilistic knowledge bases (TPKBs). TPKBs consist of a hierarchy of classes of objects and a hierarchy of classes of object pairs such that attributes and relations are independent conditioned on these classes. These characteristics facilitate both tractable probabilistic reasoning and tractable maximum-likelihood parameter learning. TPKBs feature a rich query language that allows one to express and infer complex relationships between classes, relations, objects, and their attributes. The queries are translated to sequences of operations in a relational database facilitating query execution times in the sub-second range.

We processed large data sets extracted from Wikipedia to learn a TPKB's structure and parameters. The resulting TPKB models a distribution over millions of objects and billions of parameters. We applied the TPKB to entity resolution and entity linking problems. These problem domains are important for knowledge base population and data integration, a crucial challenge for text exploration approaches. The empirical results demonstrate that the TPKB is both efficient and performs favorably compared to state-of-the-art, problem-specific and less versatile approaches. As such, TPKBs allow the user to explore and query large probabilistic knowledge bases extracted from text.

To extract knowledge bases from text, we implemented and tested a semantic parser that can infer the meaning of a sentence tractably. This parser utilizes a new theory for semantic parsing based on symmetry group theory which allows for tractable parsing without choosing a formal meaning representation or having to develop high cost training corpora. Symmetry-based semantic parsing utilizes syntactic transformations that probabilistically preserve the meaning of a sentence. The meaning of a sentence is implicitly represented by the set of sentences, called an orbit, that can be formed from each other via these meaning-preserving transformations. Due to properties of symmetry group theory, the probabilistic model of a symmetry-based semantic parser can be represented by a recursive structure of orbits over which inference can be run tractably. Structuring meaning in this way allows the semantic parser to be integrated naturally into our framework for information extraction and tractable probabilistic knowledge base construction. A symmetry-based semantic parser can be learned from a corpus of pairs of sentences with the same meaning, which is cheap and easy to create.

3.9 Symmetry-Based Paraphrase

In our work we introduced a meaning representation of text that makes use of insights from symmetry group theory [35], and can be learned in a distantly supervised manner from paraphrase corpora. Such representations abstract away syntactic variations, allowing downstream applications to focus on semantic content. For instance, it is trivial to detect paraphrases given our meaning representation since we expect sentences that are paraphrases to be mapped to the same representation. Knowledge extraction algorithms can take advantage of this to build compact and consistent knowledge bases, and to help align those knowledge bases with natural language queries [1, 8].

Our meaning representation is derived from semantic symmetry groups [23], which is based on symmetry group theory [35]. A semantic symmetry group is a set of syntactic and paraphasic transformations that preserve the meaning of a sentence. Given a single sentence, the set of sentences reachable from that sentence using these transformations is called the orbit of that sentence. An

orbit of a sentence contains sentences that have the same meaning, and the meaning representation of a sentence is the orbit it belongs to. Orbits can therefore be thought of as representing an abstract meaning by virtue of containing all sentences that express that meaning. Orbits provide a way of defining the meaning of a sentence without relying on complex, task-dependent constructs such as logical forms.

We introduced the first learning algorithm for this model of semantics. A key point to our method is taking advantage of the compositionality of natural language to efficiently represent symmetries. In particular, we allow sentences to be decomposed into constituents, where symmetries can be applied to each constituent independently. We can then represent orbits in a compact, recursive format.

In more detail, our model learns how to discover the orbit a string of text belongs to. The simplest mechanism to do so is to memorize a mapping of text to orbits, which essentially clusters text into groups that express the same meaning. However, taking this approach for phrases longer than one or two words is not scalable due to the exponentially increasing number of possible phrases. Therefore, we additionally learn rules that apply to sentences whose's constitutes belong to known orbits, allowing our model to handle longer, compositional phrases. For example, given the phrase "fast dog" our model might learn to map "fast" to an orbit including words like "quickly" and "speedy", "dog" to an orbit including "canine" and "doggy", and finally that text that consists of both those orbits in a sequence should be mapped to another orbit that captures the overall meaning of "fast dog". The resulting data structure resembles a context free grammar (CFG) where non-terminal symbols correspond to orbits instead of syntactic categories. To handle ambiguity, we extend this model to be probabilistic (analogously to how a probabilistic context free grammar extends a CFG), where we assign probabilities to each of the learned mappings.

Our learning algorithm induces how phrases (or orbit sequences) should be mapped to other orbits, and their correct probabilities. For training data we use large paraphrase corpora, which can be cheapiy produced by exploiting text translated into multiple languages [13] or parallel news streams [50]. We derived a training objective by constructing a generative paraphrase model from our meaning representation, which leverages the idea that phrases belonging from the same orbit should be paraphrases to one another. Our generative model generates a paraphrase by selecting an orbit with a learned *a-priori* probability, and then independently generating two phrases that belong to that orbit using the learned probabilities. Thus the probability of generating a particular phrase pair is:

$$P\left(phrase_1, phrase_2\right) = \sum_{i=0}^{n} P\left(o_i\right) P\left(phrase_1 | o_i\right) P\left(phrase_2 | o_i\right)$$

Where $P\left(o_i\right)$ is the *a-priori* probability of selecting a particular orbit o_i and $P\left(phrase_j | o_i\right)$ is the probability of generating $phrase_j$ from o_i. We compute $P\left(phrase_1 | o_i\right)$ using the CYK algorithm on our orbits and mappings. Our training objective then becomes the log-likelihood of the paraphrase corpus being used as training data. We additionally include a sparsity penalty on the $P\left(o_i\right)$ terms, which encourages the model use a small number of orbits.

To train the model we constructed a gradient-descent based learning algorithm. Each iteration the gradient on the probability values in our model is computed and a step is taken in that direction. We additionally perform a search for mappings, both from phrase to orbits and orbit sequences

to orbits, that have a high-gradient, and add them to the grammar with an initial probability of zero. We ensure that this search can be conducted efficiently by introducing several caching and search-space pruning tactics. We delete rules if their probability gets reduced to zero during learning. The resulting algorithm can efficiently learn the probabilities in our model and suggest discrete changes to the meaning representation.

We trained the model on the large version of the PPDB short phrase paraphrase corpus [13]. The meaning representation takes about 24 hour to learn.

We evaluated our approach on the task of paraphrase prediction. Given two phrases, we compute the meaning representation of each phrase. The meaning representations are then used to build a handful of features for that phrase-pair. The features include whether the phrases belong to the same orbit and the probability the two phrases would be generated by our generative model. We then use the features to train a linear classifier to detect if the two phrases are paraphrases. We compare this approach to using word-vector features or features derived from using several clustering algorithms on the same PPDB data. We compare on two datasets, PPDB 2.0 [38] and the AnnoPpdb dataset from [45] that was sampled to favor longer, more complex paraphrases. Our approach out-performs the clustering methods on the first dataset, and both the clustering and word-vector method on the second.

We also found that the model is able to derive a meaning representation for 55% of phrases held-out from the training data by composing learned orbit mappings. This shows the model can successfully generalize from its training data to unseen text.

Our algorithm has ready extensions to multi-lingual scenarios, where similar paraphrase data is available, and our model could be trained to map phrases in different languages to a shared meaning representation. It could also be extended to learn from plain text corpra by constructing a modified generative model that generates phrases instead of paraphrases.

4 RESULTS AND DISCUSSION

We first describe results for entity linking and then turn to event extraction.

4.1 Entity Linking

We start by describing some of the key differences amongst evaluations reported in existing literature, and propose a candidate benchmark for EL.

Data Sets Nine data sets are in common use for EL evaluation; we partition them into three groups. The UIUC group (ACE and MSNBC datasets) [40], AIDA group (with dev and test sets) [20], and TAC-KBP group (with data sets ranging from the 2009 through 2012 competitions) [32]. Their statistics are summarized in Table 5. [5]

Our set of nine is not exhaustive, but most other data sets, *e.g.* CSAW [25] and AQUAINT [36], annotate common concepts in addition to named entities. Indeed, it is extremely difficult to define annotation guidelines for common concepts, and therefore they aren't suitable for evaluation. For

[5] An online appendix containing details of the datasets is omitted to ensure blind revie

Table 5: Characteristics of the nine NEL data sets. Entity types: The AIDA data sets include named entities in four NER classes, Person (P), Organization (O), Location (L) and Misc (M). In TAC KBP data sets, both Person (P^T) and Organization entities (O^T) are defined differently from their NER counterparts and geo-political entities (G), different from L, exclude places like `KB:Central California`. KB (Sec. 4.1): The knowledge base used when each data was being developed. Evaluation Metric (Sec. 4.1): Bag-of-Concept F1 is used as the evaluation metric in [40, 3]. B3+ F1 used in TAC KBP measures the accuracy in terms of entity clusters, grouped by the mentions linked to the same entity.

Group	Data Set	# of Mentions	Entity Types	KB	# of NILs	Metric
UIUC	ACE	244	Wikipedia	Wikipedia	0	BOC F1
	MSNBC	654	Wikipedia	Wikipedia	0	BOC F1
AIDA	AIDA-dev	5917	P, O, L, M	Yago	1126	Accuracy
	AIDA-test	5616	P, O, L, M	Yago	1131	Accuracy
TACKBP	TAC09	3904	P^T,O^T,G	TAC $\subset$ Wiki	2229	Accuracy
	TAC10	2250	P^T,O^T,G	TAC $\subset$ Wiki	1230	Acc.
	TAC10T	1500	P^T,O^T,G	TAC $\subset$ Wiki	426	Acc.
	TAC11	2250	P^T,O^T,G	TAC $\subset$ Wiki	1126	B3+ F1
	TAC12	2226	P^T,O^T,G	TAC $\subset$ Wiki	1049	B3+ F1

clarity, this paper focuses on linking named entities. Similarly, we exclude datasets comprising twitter posts and other short-length documents, since radically different techniques are needed for these specialized corpora.

Table 6 presents a list of recent EL publications showing the data sets that they use for evaluation. The sparsity of this table is striking — apparently no system has reported the performance data from all three of the major evaluation groups.

Knowledge Base Existing benchmarks have also varied considerably in the knowledge base used for link targets. Wikipedia has been most commonly used [36, 40, 3], however datasets were annotated using different snapshots and subsets. Other KBs include Yago [20], Freebase [43], DBpedia [33] and a subset of Wikipedia [31]. Given that almost all KBs are descendants of Wikipedia, we use Wikipedia as the base KB in this work.[6]

NIL entities: In spite of Wikipedia's size, there are many real-world entities that are absent from the KB. When such a target is missing for a mention, it is said to link to a *NIL entity* [32] (aka out-of-KB or unlinkable entity [19]). In the TAC KBP, in addition to determining if a mention has no entity in the KB to link, all the mentions that represent the same real world entities must be clustered together. Since our focus is not to create new entities for the KB, NIL clustering is beyond the scope of this paper. We only evaluate whether a mention with no suitable entity in the KB is predicted as NIL. The AIDA data sets similarly contain such NIL annotations whereas ACE and MSNBC omit these mentions altogether.

[6]Since the knowledge bases for all the data sets were around 2011, we use a Wikipedia dump of 20110513.

Table 6: A sample of papers on entity linking with the data sets used in each paper (ordered chronologically). TAC-KBP proceedings comprise additional papers [32, 22, 22, 31]. Our intention is not to exhaust related work but to illustrate how sparse evaluation impedes comparison.

Data Set	ACE	MSNBC	AIDA-test	TAC09	TAC10	TAC11	TAC12	AQUAINT	CSAW
[6]		x							
[36]								x	
[25]		x							x
[40]	x	x						x	
[20]			x						
[16]				x					x
[17]			x		x				
[18]	x	x						x	
[3]	x	x				x		x	
[43]	x	x	x						
[28]			x	x					
[4]		x	x						x
TAC-KBP				x	x	x	x		

Evaluation Metrics While a variety of metrics have been used for evaluation, there is little agreement on which one to use. However, this detail is quite important, since the choice of metric strongly biases the results. We describe the most common metrics below.

Bag-of-Concept F1 (ACE, MSNBC): For each document, a gold bag of Wikipedia entities is evaluated against a bag of system output entities requiring exact segmentation match. This metric may have its historical reason for comparison but is in fact flawed since it will obtain 100% F1 for an annotation in which every mention is linked to the wrong entity, but the bag of entities is the same as the gold bag.

Micro Accuracy (TAC09, TAC10, TAC10T): For a list of given mentions, the metric simply measures the percentage of correctly predicted links.

TAC-KBP B3+ F1 (TAC11, TAC12): The mentions that are predicted as NIL entities are required to be clustered according to their identities (NIL clustering). The overall data set is evaluated using a entity cluster-based B3+ F1.

NER-style F1 (AIDA): Similar to official CoNLL NER F1 evaluation, a link is considered correct only if the mention matches the gold boundary *and* the linked entity is also correct. A wrong link with the correct boundary penalizes both precision and recall.

We note that Bag-of-Concept F1 is equivalent to the measure for Concept-to-Wikipedia task proposed in [4] and NER-style F1 is the same as strong annotation match. In the experiments, we use the most strict measure, NER-style F1, for evaluation over all the data sets.

Our experiments address the following questions:

- Is NER sufficient to identify mentions? (Sec. 4.1)
- How much does candidate generation affect final EL performance? (Sec. 4.1)
- How much does entity type prediction help EL? What type set is most appropriate? (Sec. 4.1)
- How much does coherence improve the EL results? (Sec. 4.1)

- Finally, how well does VINCULUM perform compared to the state-of-the-art? (Sec. 4.1)

Table 7: Recall(%) of the correct mentions using different **mention extraction** strategies.

	ACE	MSNBC	AIDA-dev	AIDA-test
NER	89.7	77.7	89.0	87.1
+NP	96.0	90.1	94.7	92.2
+DP	96.8	90.7	95.8	93.8
+NP+DP	98.0	91.9	95.9	94.1

Table 8: Performance (F1%) after **incorporating entity types**, comparing two sets of entity types (NER and FIGER). Using a set of fine-grained entity types (FIGER) generally achieves better results.

Approach	TAC09	TAC10	TAC10T	TAC11	TAC12	AIDA-dev	AIDA-test	ACE	MSNBC
CrossWikis only	80.4	85.6	86.9	78.5	62.4	62.6	60.4	87.6	82.6
+NER	79.2	83.3	85.1	76.6	61.1	66.4	**66.2**	76.8	77.9
+FIGER	**81.0**	**86.1**	**86.9**	**78.8**	**63.5**	**66.7**	64.6	**87.8**	**83.6**
+NER(GOLD)	**85.7**	87.4	88.0	80.1	**66.7**	72.6	72.0	89.2	87.1
+FIGER(GOLD)	84.1	**88.8**	**89.0**	**81.6**	66.1	**76.2**	**76.5**	**91.7**	**89.5**

Mention Extraction We start by using Stanford NER for mention extraction and measure its efficacy by the recall of correct mentions shown in Table 7. TAC data sets are not included because the mention strings are given in that competition. The results indicate that at least 10% of the gold-standard mentions are left out when NER, alone, is used to detect mentions. Some of the missing mentions are noun phrases without capitalization, a well-known limitation of automated extractors. To recover them, we experiment with an NP chunker (NP) [7] and a deterministic noun phrase extractor based on parse trees (DP). Although we expect them to introduce spurious mentions, the purpose is to estimate an upper bound for mention recall. The results confirm the intuition: both methods improve recall, but the effect on precision is prohibitive. Therefore, we only use NER in subsequent experiments.

Candidate Generation In this section, we inspect the performance of candidate generation. We compare CrossWikis with Freebase Search API [8]. Each candidate generation component takes a mention string as input and returns an ordered list of candidate entities representing the mention. We compute candidates for the union of all the non-NIL mentions from all 9 data sets and measure their efficacy by recall@k. From Figure 2, it is clear that CrossWikis consistently outperforms Freebase Search API for all k.

Using CrossWikis for candidate generation, we plot the recall@k curves per data set (Figure 3). To our surprise, in most data sets, CrossWikis alone can achieve more than 70% recall@1.

---

[7] OpenNLP NP Chunker: `opennlp.apache.org`

[8] `https://www.googleapis.com/freebase/v1/search`, restricted to no more than 220 candidates per query.

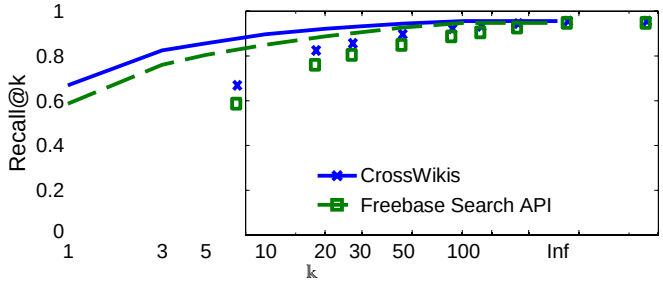

Figure 2: Recall@k on an aggregate of nine data sets, comparing two **candidate generation** methods.

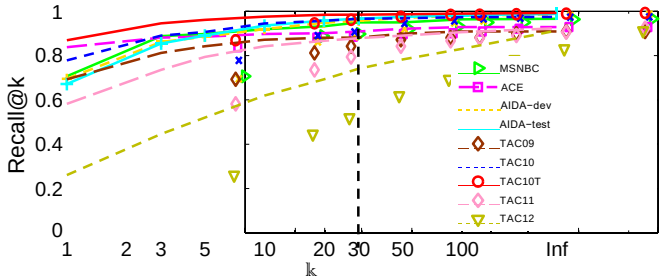

Figure 3: Recall@k using CrossWikis for candidate generation, split by data set. 30 is chosen to be the cut-off value in consideration of both efficiency and accuracy.

The only exceptions are TAC11 and TAC12 because the organizers intentionally selected the mentions that are highly ambiguous such as "ABC" and/or incomplete such as "Brown". For efficiency, we set a cut-off threshold at 30 (> 80% precision for all but one data set).

Incorporating Entity Types Here we investigate the impact of the entity types on the linking performance. The most obvious choice is the traditional NER types ($T_{\text{NER}} = \{$PER, ORG, LOC, MISC$\}$. To predict the types of the mentions, we run Stanford NER [12] and set the predicted type t_m of each mention m to have probability 1 (*i.e.* $p(t_m | m, s) = 1$). As to the types of the entities, we map their Freebase types to the four NER types[9].

A more appropriate choice is 112 fine-grained entity types introduced by [29] in FIGER, a publicly available package [10]. These fine-grained types are not disjoint, i.e. each mention is allowed to have more than one type. For each mention, FIGER returns a set of types, each of which is accompanied by a score, $t_{\text{FIGER}}(m) = \{(t_j, g_j) : t_j \in T_{\text{FIGER}}\}$. A softmax function is used to probabilistically interpret the results as follows:

$$p(t_j|m,s) = \begin{cases} \frac{1}{Z}\exp(g_j) & \text{if } (t_j, g_j) \in t_{\text{FIGER}}(m), \\ 0 & \text{otherwise} \end{cases}$$

We evaluate the utility of entity types in Table 8, which shows that using NER typically worsens the performance. This drop may be attributed to the rigid binary values for type incorporation; it is hard to output the probabilities of the entity types for a mention given the chain model adopted in Stanford NER. We also notice that FIGER types consistently improve the results across the data sets, indicating that a finer-grained type set may be more suitable for the entity linking task.

To further confirm this assertion, we simulate the scenario where the gold types are provided for each mention (the oracle types of its gold entity). The performance is significantly boosted with the assistance from the gold types, which suggests that a better performing NER/FIGER system can further improve performance. Similarly, we notice that the results using FIGER types almost consistently outperform the ones using NER types. This observation endorses our previous recommendation of using fine-grained types for EL tasks.

[9]The Freebase types "/person/*" are mapped to PER, "/location/*" to LOC, "/organization/*" plus a few others like "/sports/sports team" to ORG, and the rest to MISC.

[10] github.com/xiaoling/fige

Table 9: Performance (F1%) after re-ranking candidates using coherence scores, comparing two **coherence measures** (NGD and REL). "no COH": no coherence based re-ranking is used. "+BOTH": an average of two scores is used for re-ranking. Coherence in general helps: a combination of both measures often achieves the best effect and NGD has a slight advantage over REL.

Approach	TAC09	TAC10	TAC10T	TAC11	TAC12	AIDA-dev	AIDA-test	ACE	MSNBC
no COH	80.9	86.2	87.0	78.6	59.9	68.9	66.3	87.8	86.2
+NGD	**81.8**	85.7	86.8	79.7	63.2	**69.5**	**67.7**	88.0	**86.7**
+REL	81.2	86.3	87.0	79.3	63.1	69.1	66.4	**88.4**	86.6
+BOTH	81.4	**86.8**	**87.0**	**79.9**	**63.7**	69.4	67.5	**88.4**	**86.7**

Table 10: **End-to-end performance**: We compare VINCULUM in different stages with two state-of-the-art systems, AIDA and WIKIFIER, in F1(%). The column "Overall" lists the average performance of nine data sets for each approach. CrossWikis appears to be a strong baseline. VINCULUM is 0.4% shy from WIKIFIER, each winning in four data sets; AIDA tops both VINCULUM and WIKIFIER on AIDA-test.

Approach	TAC09	TAC10	TAC10T	TAC11	TAC12	AIDA-dev	AIDA-test	ACE	MSNBC	Overall
CrossWikis	80.4	85.6	86.9	78.5	62.4	62.6	62.4	87.6	82.6	76.3
+FIGER	81.0	86.1	86.9	78.8	63.5	66.7	64.5	87.8	83.6	77.7
+Coref	80.9	86.2	87.0	78.6	59.9	68.9	66.3	87.8	86.2	78.0
+Coherence =VINCULUM	**81.4**	**86.8**	**87.0**	79.9	63.7	69.4	67.5	**88.4**	86.7	79.0
AIDA	73.2	78.6	77.5	68.4	52.0	71.9	**74.8**	77.8	75.4	72.2
WIKIFIER	79.7	86.2	86.3	**82.4**	**64.7**	**72.1**	69.8	85.3	**88.2**	**79.4**

Coherence Two coherence measures suggested in Section 3.1 are tested in isolation to better understand their effects in terms of the linking performance (Table 9). In general, the link-based NGD works slightly better than the relational facts in 6 out of 9 data sets (comparing row "+NGD" with row "+REL"). We hypothesize that the inferior results of REL may be due to the incompleteness of Freebase triples, which makes it less robust than NGD. We also combine the two by taking the average score, which in most data set performs the best ("+BOTH"), indicating that two measures provide complementary source of information.

Overall Performance Analysis To answer the last question of how well does VINCULUM perform overall, we conduct an end-to-end comparison against two publicly available systems with leading performance:[11]

AIDA [20]: We use the recommended GRAPH variant of the AIDA package and are able to replicate their results when gold-standard mentions are given.

WIKIFIER [3]: We are able to reproduce the reported results on ACE and MSNBC and obtain a close enough B3+ F1 number on TAC11 (82.4% vs 83.7%). Since WIKIFIER overgenerates mentions and produce links for common concepts, we restrict its output on the AIDA data to the mentions that Stanford NER predicts.

Table 10 shows the performance of VINCULUM after each stage of candidate generation (CrossWikis), entity type prediction (+FIGER), coreference (+Coref) and coherence (+Coherence). The column "Overall" displays the average of the F1 numbers for nine data sets for each approach. WIKIFIER achieves the highest in the overall performance. VINCULUM performs quite comparably, only 0.4% shy from WIKIFIER, despite its simplicity and unsupervised nature. Looking at the performance per data set, VINCULUM and WIKIFIER each is superior in 4 out of 9 data sets while AIDA tops the performance only on AIDA-test.

We notice that even using CrossWikis alone works pretty well, indicating a strong baseline for future comparisons. The entity type prediction provides the highest boost on performance, an absolute 1.4% increase, among other subcomponents. The coherence stage also gives a reasonable lift. Last but not the least, VINCULUM runs reasonably fast. For a document with 20-40 entity mentions on average, VINCULUM takes only a few seconds to finish the linking process on one single thread.

4.2 Event Extraction

UW submitted five runs for the English EAL task. In this task, teams were presented with a corpus of 30,002 documents, evenly divided between newswire and discussion forum text, with the goal of extracting arguments participating in a set of 18 event subtypes, identifying both event subtype and role. Both Evento and NomEvent were trained on the ACE 2005 corpus, with NomEvent's training corpus supplemented with Rich ERE data from LDC2016E60.

Run Washington1 aimed for maximum recall by combining the Evento and NewsSpike systems with the NomEvent system trained to classify all parts of speech. The union of the events returned by the three systems was used, with the Evento result chosen when overlapping argument extractions disagreed on the extent or role of the argument.

Run Washington2 was identical to Washington1 but substituted the NomEvent system trained to only extract nominal events.

Run Washington3 consisted solely of the Evento system.

Run Washington4 consisted solely of the NomEvent system, classifying all potential triggers.

The final run, Washington5, aimed for a high-precision result by considering the results returned by the Evento, NewsSpike, and NomEvent (all-part-of-speech) systems and keeping only results returned by at least two of the three systems.

[11] We are also aware of other systems such as TagMe-2 [11], DBpedia Spotlight [33] and WikipediaMiner [36]. A trial test on the AIDA data set shows that both Wikifier and AIDA tops the performance of other systems reported in [4] and therefore it is sufficient to compare with these two systems in the evaluation.

Table 11: Scores for the Event Argument Extraction and Linking evaluation.

TAC KBP 2016 EAL Evaluation Results								
System	TP	FP	FN	ArgP	ArgR	ArgF1	ArgScore	LinkScore
Wash1	440	1223	6065	26.5	**6.8**	**10.8**	3.2	**2.0**
Wash2	343	948	6162	26.6	5.3	8.8	2.6	1.3
Wash3	247	717	6258	25.6	3.8	6.6	2.0	0.7
Wash4	327	691	6178	32.1	5.0	8.7	**3.3**	1.5
Wash5	120	144	6385	**45.5**	1.8	3.5	1.6	0.3

Table 12: Results for the 2016 Event Nugget Detection evaluation.

TAC KBP 2016 Event Nugget Evaluation Results							
		Micro			Macro		
System	Attributes	Prec	Rec	F1	Prec	Rec	F1
Washington1	plain	50.19	**35.02**	**41.25**	47.34	**33.11**	**38.97**
	mention_type	42.15	**29.41**	**34.65**	38.95	**27.50**	**32.24**
	realis_status	36.20	**25.25**	**29.75**	34.18	**23.58**	**27.91**
	mention_type+realis_status	30.71	**21.42**	**25.24**	28.35	**19.75**	**23.28**
Washington2	plain	49.76	33.01	39.69	47.14	31.09	37.47
	mention_type	41.83	27.75	33.36	38.68	25.79	30.95
	realis_status	36.38	24.13	29.02	34.61	22.66	27.39
	mention_type+realis_status	30.97	20.55	24.70	28.78	19.04	22.92
Washington3	plain	**62.15**	26.64	37.29	**57.12**	24.54	34.33
	mention_type	**55.96**	23.99	33.58	**50.89**	21.97	30.69
	realis_status	**45.22**	19.38	27.14	**40.75**	17.66	24.64
	mention_type+realis_status	**41.10**	17.62	24.66	**36.61**	15.92	22.19

The 2016 English Event Nugget task provided a corpus of 169 documents, split between newswire and discussion forum text and required teams to extract event triggers corresponding to the same ontology of 18 events used in EAL. UW submitted three runs for this task.

The first run, Washington1, consisted of the union of an Evento run tuned for F1, a NomEvent run trained on all parts of speech, and a NewsSpike run. Run Washington2 consisted of the union of an Evento run tuned for F1, a NomEvent run trained on nominal events, and a NewsSpike run. Washington3 was identical to Washington1 but was tuned for high precision.

Table 11 shows detailed results of the EAL evaluation, in which UW scored above the median for both the Argument and Linking scores. The median Argument score over the top performing system from each team was 3.0; Washington4 topped this with a 3.3, as did Washington1 with a

3.2. For the linking score, Washington1 posted a 2.0, above the median of 1.6. As anticipated, Washington1 posted the highest recall of the UW systems, while Washington5 posted the highest precision.

Table 12 shows detailed results of the event nugget evaluation. Washington1 posted higher recall

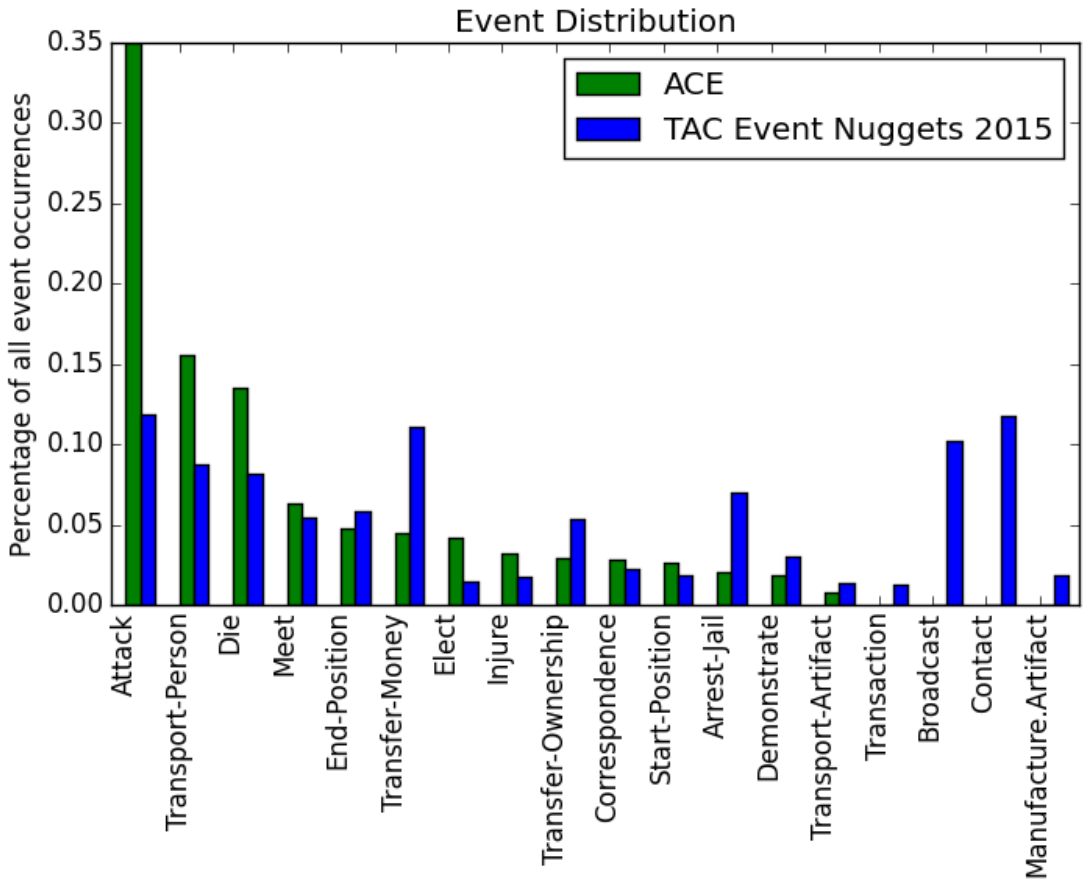

Figure 4: Comparison of event distribution in ACE 2005 dataset as compared to the TAC KBP Event Nugget 2015 task.

and F1 scores than the other UW systems, while Washington3 turned in the highest precision, as expected. An examination of the event breakdown revealed that Washington1 earned its highest F1 scores on the Life.Injure (0.64), Life.Die (0.54), and Justice.ArrestJail (0.60) events, while it struggled on Contact.Contact (0.01), Contact.Broadcast (0.01), and Transaction.Transaction (0.0),
Manufacture.Artifact (0.0) and Movement.TransportArtifact (0.0).

Failure Analysis The poor performance on some events is likely due to the distribution of events in the test data as compared to the ACE 2005 corpus which provided the bulk of the training data for the Evento and NomEvent systems, as shown in Figure 1. These five events made up almost

20% of events in this year's Event Nugget evaluation, so training on more representative data would likely have significantly boosted the performance of our systems.

Evento places a lot of weight on the large number of lexical features it uses. Because of this, it generalizes poorly to triggers that do not appear in the training set, especially when they appear with new contexts that also were not seen in training. This results in low recall for events with a diverse set of triggers, such as *start-org*, *transfer-ownership*, and *transfer-money*.

A failure analysis of NomEvent discovered that 58% of false positives corresponded to a misclassification of closely related events, such as classifying a Contact.Broadcast event as Contact.Correspondence. The vast majority of missed triggers (88%) were words which were not in the generated lexicon.

We tested our semi-supervised method using two labeled datasets: ACE-2005 and TAC-KBP 2015. For the ACE data, we use the same train/development/test split as has been previously used in [27], consisting of 529 training documents, 30 development documents, and a test set consisting of 40 newswire articles containing 672 sentences. For the TAC-KBP 2015 dataset, we use the official train/test split as previously used in [39] consisting of 158 training documents and 202 test documents. ACE contains 33 event types, and TAC-KBP contains 38 event types.

For our approach, we use a collection of news articles scraped from the web. These articles were scraped following the approach described in [51]. The process involves collecting article titles from RSS news seeds, and then querying the Bing news search with these titles to collect additional articles. This process was repeated on a daily basis between January 2013 and February 2015, resulting in approximately 70 million sentences from 8 million articles. Although the seed titles were collected during that two year period, the search results include articles from prior years with similar titles, so the articles range from 1970 to 2015.

Evaluation We report the micro-averaged F1 scores over all events. A trigger is considered correctly labeled if both its offsets and event type match those of a reference trigger.

For creating the automatically-generated data, we set thresholds θ_{event} and θ_{sim} to 2 and 0.4 respectively, which were selected according to validation data. We use CoreNLP [30] for named entity recognition, and we use a pre-trained Word2Vec model [34] for the vector representations.

For the JRNN model, we follow the parameter settings of [37] and use a context window of 2 for context words, and a feed-forward neural network with one hidden layer for trigger prediction with hidden layer size of 300. Finally, for training, We apply the stochastic gradient descent algorithm with mini-batches of size 50 and the AdaDelta update rule [47] with L_2 regularization. For the CRF model, we maximize the conditional log likelihood of the training data with a loss function via softmax-margin [14]. We optimize using AdaGrad [7] with L_2 regularization.

Varying Amounts of Additional Data In this section we show that the addition of automatically-generated training examples improves the performance of both systems we tested it on. We sample examples from the automatically-generated data, limiting the total number of positive examples to a specific number. In order to avoid biasing the system in favor of a specific event type, we ensure that the additional data has a uniform distribution of event types. We run 10 trials at each point, and report average results.

Table 13 reports the results of adding varying amounts of our generated data to both CRF and JRNN systems. We observe that that adding any amount of heuristically-generated data improves performance. Optimal performance, however, is achieved fairly early in both datasets. This is likely

due to the domain mismatch between the gold and additional data. For reference purposes, we also include the result of using the HNN model from [9] and the SSED system from [42], which are the best reported results on the ACE-2005 and TAC-KBP 2015 corpora respectively.

Table 13: Results after adding varying amounts of automatically-generated news data. Percentages indicate the amount of additional data relative to the size of the gold training data. Using a modest amount of semi-supervised data improves extractor performance on both ACE & TAC-KBP events. * indicates that the difference in F1 relative to training with just the gold data is statistically significant ($p < 0.05$).

		ACE			TAC-KBP		
		P	R	F1	P	R	F1
CRF	0%	62.9	70.0	66.3	53.5	52.3	52.9
	10%	64.5	69.8	67.0	**59.9**	49.3	**54.1***
	20%	**65.1**	**70.2**	**67.6***	59.3	49.2	53.8
	30%	**65.1**	69.9	67.4	58.1	**49.4**	53.4
JRNN	0%	65.7	72.9	69.1	**68.8**	49.2	57.3
	10%	67.4	72.7	69.9	65.4	52.1	58.0
	20%	**67.6**	**73.5**	**70.4***	65.3	52.8	**58.4***
	30%	67.5	73.3	70.3	64.7	**52.9**	58.2
HNN		84.6	64.9	73.4	-	-	-
SSED		-	-	-	69.9	48.8	57.5

These systems could also benefit from our additional data since our approach is system independent.

Varying Amounts of Supervised Data In this section we evaluate how the benefit of adding semi-supervised data varies given different amounts of gold (supervised) data to start. We conjecture that semi-supervision will be more beneficial when gold data is very limited, but the conclusion isn't obvious, since semi-supervision is more likely to add noisy examples in this case. Specifically, we limit the number of positive gold examples for each event by randomly sampling the overall set. We then add in the same amount of automatically-generated data to each trial. We again run 10 trials for each size, and report the average.

The results for this experiment using the CRF model can be seen in figure 5: training with large amounts of semi-supervised data improves performance considerably when limited gold training data is available, but those gains diminish with more high-quality supervised data. We observe the same trend for the JRNN system as well.

Discussion We randomly selected 100 examples from the automatically-generated data and manually annotated them. For each example that did not contain a correctly labeled event mention,
we further annotated where in the pipeline an error occurred to cause the incorrect labeling. This breakdown can be seen in table 14. As observed in the table, the errors are mainly due to the incorrect event identification or trigger assignment.

Incorrect clustering refers to cases in which a sentence does not cover the same topic as other sentences in its cluster. This was primarily caused by entities participating in multiple events around the same time period. For example, this occurred in sentences from the 2012 US presidential election coverage involving Barack Obama and Mitt Romney.

Incorrect event identification refers to clusters that were incorrectly labeled by the supervised system. The primary reason for these errors is due to domain mismatch between the news articles and the gold data. For example, our system identifies the token *shot* in *Bubba Watson shot a 67 on Friday* as an attack event trigger. Because the gold data does not contain examples involving sports, the baseline system mistakenly identifies a paraphrase of the above sentence as an attack event, and our system is not able to fix that mistake. However, this problem can be solved by training the baseline extractor on the same domain as the additional data.

Incorrect trigger assignment refers to errors in which a sentence is correctly identified as containing an event mention, but the wrong token is selected as a trigger. The most common source of this error is tokens that are strongly associated with multiple events. For example, *shooting* is strongly associated with both attack and die events, but only actually indicates an attack event.

Looking through the correct examples, the data collection process is able to identify uncommon triggers that do not show up in the baseline training data. For example, it correctly identifies "offload" as a trigger for Transfer-Ownership in *Barclays is to offload part of its Spanish business to Caixabank*. Despite the trigger identification step having no context awareness, the process is also

able to correctly identify triggers that rely on context, such as "contributions" triggering Transfer- Money in *Chatwal made $188,000 of illegal campaign contributions to three U.S. candidates via straw donors*.

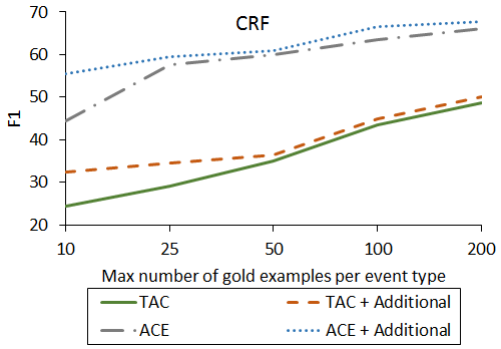

Figure 5: Adding a reasonable amount (200 examples per event) of semi-supervised data on top of limited amounts of gold training data improves performance across the board, but the gain is dramatic when the number of supervised examples is extremely small.

Table 14: The results of manually labeling mples that were automatically-generated using JRNN as the supervised system.

Correct		72
Incorrect	clustering	5
	event identification	13
	trigger assignment	10

5 CONCLUSIONS

We have made significant progress on several information extraction technologies.

We present a simple yet effective, modular, unsupervised system, VINCULUM, for entity linking. We make the implementation open source and publicly available for future research. We compare VINCULUM to 2 state-of-the-art systems on an extensive evaluation of 9 data sets. We also investigate several key aspects of the system including mention extraction, candidate generation, entity type prediction, entity coreference, and coherence between entities.

We also developed new algorithms for event extraction that exploit parallel news streams. These methods cluster sentences that belong to the same event relations using the temporal negation heuristic and a novel probabilistic graphical model to generate training for a sentential event extractor without requiring any human annotations. We also introduce a semi-supervised method for combining labeled and unlabeled data for event extraction, showing significant performance improvements on multiple event extractors over ACE 2005 and TAC-KBP 2015 datasets.

In addition to our core work on IE, we developed new natural language processing tools (semantic parsing) and worked on efficient algorithms for inference over extracted knowledge bases.

6 REFERENCE

S References

[1] Jonathan Berant and Percy Liang. Semantic Parsing via Paraphrasing. In *ACL*, 2014.

[2] Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL*, 2015.

[3] Xiao Cheng and Dan Roth. Relational inference for wikification. In *EMNLP*, 2013.

[4] Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd international conference on World Wide*

Web, pages 249–260. International World Wide Web Conferences Steering Committee, 2013.

[5] Mark Craven and Johan Kumlien. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 77–86, 1999.

[6] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of EMNLP-CoNLL*, volume 2007, pages 708–716, 2007.

[7] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *Journal of Machine Learning Research*, 2011.

[8] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Paraphrase-Driven Learning for Open Question Answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013.

[9] Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. A language-independent neural network for event detection. In *ACL*, 2016.

[10] James Ferguson, Colin Lockard, Natalie Hawkins, Stephen Soderland, Hannaneh Hajishirzi, and Daniel S. Weld. University of washington tac-kbp 2016 system description. In *TAC-KBP*, 2017.

[11] Paolo Ferragina and Ugo Scaiella. Fast and accurate annotation of short texts with wikipedia pages. *IEEE Software*, 29(1):70–75, 2012.

[12] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[13] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. PPDB: The Paraphrase Database. In *HLT-NAACL*, 2013.

[14] Kevin Gimpel and Noah A. Smith. Softmaxmargin crfs: Training log-linear models with cost functions. In *HLT-NAACL*, 2010.

[15] Hannaneh Hajishirzi, Leila Zilles, Daniel S. Weld, and Luke Zettlemoyer. Joint Coreference Resolution and Named-Entity Linking with Multi-pass Sieves. In *EMNLP*, 2013.

[16] Xianpei Han and Le Sun. An entity-topic model for entity linking. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 105–115. Association for Computational Linguistics, 2012.

[17] Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. Learning entity representation for entity disambiguation. *Proc. ACL2013*, 2013.

[18] Zhengyan He, Shujie Liu, Yang Song, Mu Li, Ming Zhou, and Houfeng Wang. Efficient collective entity linking with stacking. In *EMNLP*, pages 426–435, 2013.

[19] Johannes Hoffart, Yasemin Altun, and Gerhard Weikum. Discovering emerging entities with ambiguous names. In *Proceedings of the 23rd international conference on World wide web*, pages 385–396. International World Wide Web Conferences Steering Committee, 2014.

[20] Johannes Hoffart, Mohamed A. Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics, 2011.

[21] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT-ACL)*, pages 541–550, 2011.

[22] Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. Overview of the tac 2010 knowledge base population track. In *Text Analysis Conference (TAC 2010)*, 2010.

[23] Chloe´ Kiddon and Pedro Domingos. Symmetry-Based Semantic Parsing. 2014.

[24] Mitchell Koch, John Gilmer, Stephen Soderland, and Daniel S Weld. Type-aware distantly supervised relation extraction with linked arguments. In *EMNLP*, 2014.

[25] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of Wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466. ACM, 2009.

[26] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, pages 1–54, 2013.

[27] Qi Li, Heng Ji, and Liang Huang. Joint event extraction via structured prediction with global features. In *ACL*, 2013.

[28] Yang Li, Chi Wang, Fangqiu Han, Jiawei Han, Dan Roth, and Xifeng Yan. Mining evidences for named entity disambiguation. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1070–1078. ACM, 2013.

[29] Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2012.

[30] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In

Association for Computational Linguistics (ACL) System Demonstrations, pages 55–60, 2014.

[31] James Mayfield, Javier Artiles, and Hoa Trang Dang. Overview of the tac2012 knowledge base population track. *Text Analysis Conference (TAC 2012)*, 2012.

[32] P. McNamee and H.T. Dang. Overview of the tac 2009 knowledge base population track. *Text Analysis Conference (TAC 2009)*, 2009.

[33] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.

[34] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[35] Willard Miller. *Symmetry Groups and their Applications*. 1973.

[36] David Milne and Ian H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM, 2008.

[37] Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. Joint event extraction via recurrent neural networks. In *HLT-NAACL*, 2016.

[38] Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison- Burch. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, wordembed- dings, and style classification. 2015.

[39] Haoruo Peng, Yangqiu Song, and Dan Roth. Event detection and co-reference with minimal supervision. In *EMNLP*, 2016.

[40] Lev-Arie Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *ACL*, volume 11, pages 1375–1384, 2011.

[41] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases (ECML)*, pages 148–163. Springer, 2010.

[42] Mark Sammons, Haoruo Peng, Yangqiu Song, Shyam Upadhyay, Chen-Tse Tsai, Pavankumar Reddy, Subhro Roy, and Dan Roth. Illinois ccg tac 2015 event nugget, entity discovery and linking, and slot filler validation systems. In *TAC*, 2015.

[43] Avirup Sil and Alexander Yates. Re-ranking for joint named-entity recognition and linking. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2369–2374. ACM, 2013.

[44] Valentin I Spitkovsky and Angel X Chang. A cross-lingual dictionary for english wikipedia concepts. In *LREC*, pages 3168–3175, 2012.

[45] John Wieting, Mohit Bansal, Kevin Gimpel, Karen Livescu, and Dan Roth. From Paraphrase Database to Compositional Paraphrase Model and Back. *arXiv preprint arXiv:1506.03487*, 2015.

[46] Fei Wu and Daniel S. Weld. Autonomously semantifying wikipedia. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 41–50, 2007.

[47] Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

[48] Congle Zhang, Stephen Soderland, and Daniel S. Weld. Exploiting parallel news streams for unsupervised event extraction. *TACL*, 3:117–129, 2015.

[49] Congle Zhang and Daniel S Weld. Harvesting parallel news streams to generate paraphrases of event relations. In *Proceedings of the 2013 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP)*, pages 455–465, 2013.

[50] Congle Zhang and Daniel S Weld. Harvesting Parallel News Streams to Generate Paraphrases of Event Relations. In *EMNLP*, 2013.

[51] Congle Zhang and Daniel S. Weld. Harvesting parallel news streams to generate paraphrases of event relations. In *EMNLP*, 2013.

[52] Jiaping Zheng, Luke Vilnis, Sameer Singh, Jinho D. Choi, and Andrew McCallum. Dynamic knowledge-base alignment for coreference resolution. In *Conference on Computational Natural Language Learning (CoNLL)*, 2013.