# UI_CCG TAC-KBP2017 Submissions: Entity Discovery and Linking, and Event Nugget Detection and Co-reference

**Chase Duncan**[1], **Liang-Wei Chan**[1], **Haoruo Peng**[1], **Hao Wu**[2],
**Shyam Upadhyay**[2], **Nitish Gupta**[2], **Chen-Tse Tsai**[1]
**Mark Sammons**[1], **Dan Roth**[1,2]
[1]University of Illinois at Urbana-Champaign, [2]University of Pennsylvania

## 1 Introduction

The UI_CCG team participated in the Entity Discovery and Linking (EDL), Event Nugget Detection, and Event Nugget Co-reference tasks.

The EDL system extended Illinois Cross Lingual Wikifier (Tsai and Roth, 2016; Tsai et al., 2016b), with nominal detection and linking and a module for enforcing coherent assignments of links across the entity mentions. The system was developed for all three evaluation languages: English, Chinese and Spanish.

The system for the Event Nugget tasks built on work that UI_CCG submitted to prior TAC evaluations (Tsai et al., 2016a). UI_CCG submitted runs for English and Spanish. For English, the system employs supervised models with rich lexical and semantic features; while for Spanish, we use Google Translation to convert text into English so we can use the English-trained system, then map the event output back to the original text.

## 2 Entity Detection, Classification and Linking

The UI_CCG Entity Detection and Linking system is based on the Illinois Cross Lingual Wikifier (Tsai and Roth, 2016; Tsai et al., 2016b). There are separate sub-systems for Named Entity mentions and for non-proper-noun Nominal mentions. Each sub-system has separate mention extraction and linking steps. The Named Entity sub-system also has a NIL clustering step. Table 1 shows the system performance.

### 2.1 Named Entity Detection and Linking

**Named Mention Extraction:** The Illinois Cross Lingual Wikifier (XLWikifier) extends the publicly available Illinois Named Entity Recognition (NER) (Ratinov and Roth, 2009; Redman et al., 2016) system to detect named entities in the cross lingual setting. The cross-lingual NER is language independent, leveraging wikification to yield language-independent features based on Wikipedia categories and Freebase types. Although the main idea in Tsai et al. (2016b) is to train a model on one language and apply it on another language directly, the authors also show that the newly proposed wikifier features are useful in monolingual models.

For the training data, we use TAC EDL 2015 training and evaluation documents, TAC EDL 2016 evaluation documents, and the ERE datasets. The model is trained on the training data for each language, therefore there is one monolingual model per language.

**Entity Linking:** The next step is to ground the extracted named entity mentions to the English Wikipedia. We apply the model proposed in Tsai and Roth (2016) which uses cross-lingual word and title embeddings to compute similarities between a foreign mention and English title candidates. We then obtain the corresponding FreeBase ID using the links between Wikipedia titles and FreeBase entries if a mention is grounded to some Wikipedia entry.

**NIL Clustering:** For the named entity mentions which could not be grounded to the knowledge base, we try to group them with other named entities by the NIL clustering algorithm which we developed in TAC EDL 2015 (Sammons et al., 2015). The initial clustering is based on the Wikification result, where each NIL mention forms a singleton cluster. These initial clusters are sorted by their size. We merge clusters greedily: a smaller cluster will be merged into a larger cluster if there is any pair of mentions from two different clusters that are sufficiently similar. The similarity between two mentions is based on the Jaccard similarity of the surface strings.

| 2017 Evaluation Set | | | |
|---|---|---|---|
| Measure | Precision | Recall | F1 |
| Trilingual | | | |
| strong typed mention match | 85.2 | 65.3 | 73.9 |
| strong typed all match | 68.4 | 52.4 | 59.3 |
| typed mention ceaf plus | 58.3 | 44.6 | 50.5 |
| English | | | |
| strong typed mention match | 86.1 | 72.7 | 78.9 |
| strong typed all match | 67.7 | 57.2 | 62.0 |
| typed mention ceaf plus | 60.8 | 51.4 | 55.7 |
| Spanish | | | |
| strong typed mention match | 85.7 | 68.3 | 76.0 |
| strong typed all match | 65.7 | 52.4 | 58.3 |
| typed mention ceaf plus | 61.7 | 49.1 | 54.7 |
| Chinese | | | |
| strong typed mention match | 84.1 | 58.1 | 68.7 |
| strong typed all match | 71.2 | 49.2 | 58.2 |
| typed mention ceaf plus | 68.8 | 47.5 | 56.2 |

Table 1: EDL System performance on TAC-KBP 2017 evaluation.

### 2.2 Nominal Mention Detection

The nominal mention detection task requires the EDL system to identify referring phrases that are not proper nouns, and link them to target referents. We refer to these phrases as "nominal mentions". Here we introduce a supervised learning framework that trains a nominal mention detector from annotated data.

#### 2.2.1 Model

We use the same regularized averaged perceptron algorithm that has been shown to be competitive in NER and text chunking (Ratinov and Roth, 2009). It uses expressive features to achieve the state-of-the-art performance on the named entity recognition task.

For the representation of mention extents, we use the most popular schema - BIO. The BIO schema classifies each token as either the **B**eginning, the **I**nside, or the **O**utside of a semantic chunk (in this task, a nominal referring phrase). Given the chunk labels for the task, this results in 11 classification categories: {FAC, GPE, LOC, ORG, PRE} X {B, I} + {O}. Though our model can represent mentions of any length, for English and Spanish the mentions annotated for the TAC EDL task are usually length 1, so most of the positive tags are B. However, mentions in Chinese text are usually longer, so there is a greater proportion of I-tags.

#### 2.2.2 Features

The algorithm considers each word from the text in order (the "focus word"), and features are extracted for each. We use a subset of the features used for NER in (Ratinov and Roth, 2009), and characterize them in terms of three types:

1. **Lexical Features**: Lexical features are the features that are extracted from the surface form of a word.
   (a) Forms: the surface form of a word and its neighbor words. This kind of feature is essential for nominal detection because many of the nominals appear multiple times with the same surface form.
   (b) Capitalization: whether the word is capitalized. This feature is useful to filter out modifiers for languages such as English, e.g., "president Obama", though it does not help for Chinese.
   (c) Word Type Information: whether the word is composed of lowercase letters, uppercase letters, or digits. In the EDL task, nominals cannot be purely numerical expressions.
   (d) Affixes: the prefix and the suffix of a word. Suffixes are effective for identifying common nominals – for example, the suffix "-ist" in the words "scientist" or "artist".
   (e) Previous tag pattern: the surface form of the current word, the surface form of the previous word, and the tag predicted for that word.

2. **Non-lexical Features**: Non-lexical features go beyond the surface form of the focus word.
   (a) Previous tags: the predicted tag of the word before focus word. In Chinese, this feature is vital because for I- tags, the previous tag must be a B- tag.
   (b) Tag context: Given the word we are predicting and the next two words, we use statistics for tag sequences for word trigrams and their predicted tags over the previous 1000 words in the document.

3. External Resources: We also use Brown clusters features that are obtained from the hierarchical clustering of words based on the contexts they occur in. These are the same as those used in (Tsai et al., 2016b).

#### 2.2.3 Post-Processing with Dictionaries

The set of nominals is smaller than that of named entities. However, the model is prone to overfitting when the amount of training data is small. As a result, we introduce a post-processing step to relieve the potential overfitting phenomenon. We build a dictionary from the gold annotations in the training data with the top N frequent words. When testing, if we encounter words that are in the dictionary that were not detected by the model, we annotate them as nominal head mentions.

For example, given that we have seen "government" in the gold annotations with high frequency, we put it into the dictionary. When we encounter a sentence "The government states ...", we then detect the word government as a head mention.

#### 2.2.4 Performance

Table 2 shows the precision, recall, and F1-score of the nominal detector on the 2017 evaluation data. The English and Spanish have similar performance, which is around 60% F1-score. However, for Chinese, it turns out that the nominal detection is much more difficult.

Table 3 shows the change of performance when we ablate the post-processing step, which results in a trade-off between precision and recall. Initially, our nominal mention detection models have higher precision but lower recall. Adding the top N frequent nominals into the dictionary increases the recall but hurts the precision because some of the added nominals are not always valid in context. In English, the F-1 score does not improve much because the precision and recall are already close before using the dictionaries. For Spanish and Chinese, however, the dictionary-based post-processing helps to reduce the gap between precision and recall which is larger, resulting in increased F-1 .

### 2.3 Nominal Linking

This section describes the process for linking the detected nominal heads to the corresponding entities. For example, given the following short paragraph:

*Apple released new details about iPhone X. The company moved to speed up production as pre-orders loom.*

the nominal "company" should be linked to knowledge base entry **Apple Inc.**; the correct link can be determined via co-reference resolution. However, sometimes the linking may not involve co-reference resolution. Consider the following example:

*China is the world's most populous country. Its government aimed to control the population growth.*

Here the named entity "China" refers to the country **the People's Republic of China** while the nominal "government" should be linked to **Government of China** and thus there is no co-reference information. In the case where the target knowledge base entry doesn't exist, we should link the nominal to NIL (non-linkable), which is also called the NIL identification task. For instance, consider:

*My friend told me an interesting story yesterday.*

Here the nominal "friend" has type **PER**, but "my friend" may not have a Wikipedia page.

In summary, a nominal mention can be (1) co-referable and linkable; (2) co-referable and non-linkable; (3) non-co-referable and linkable; or (4) non-co-referable and non-linkable.

#### 2.3.1 Linking algorithms

In this section, we introduce heuristics that are helpful for nominal linking. Given the linked named entities and the discovered nominal mentions, we propose the following algorithms.

1. **Nearest Typed-based Co-reference:** One naive algorithm for nominal linking uses type and proximity. For each nominal mention *m*, we identify the nearest mention that has the same type as *m* and mark them as co-referring. Nearest Type-based Co-reference is a strong baseline for nominal linking, especially for mentions of type **PER**.

2. **Co-reference by Substring Matching of Freebase Types:** We can gather useful information from the freebase fine-grained types. For example, Kiev is the capital city of Ukraine, and one of its freebase types is "independent_city". Given a nominal "city", we should check whether the nominal matches the type. The algorithm is as follows:

   (a) Given a Co-reference candidate named entity, get fine-grained types from its Freebase MID.

   (b) Check whether the nominal is the substring of any freebase type.

   (c) Return the nearest validated named entity.

| | # Mentions | Mention Span Matches | | | (Mention Span + Type) Matches | | |
|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1-Score | Precision | Recall | F1-score |
| English | 1726 | 69.91 | 56.55 | 62.52 | 66.33 | 53.65 | 59.32 |
| Spanish | 1915 | 75.13 | 53.00 | 62.16 | 70.76 | 49.92 | 58.54 |
| Chinese | 2348 | 49.41 | 26.58 | 34.56 | 47.43 | 25.51 | 33.18 |

Table 2: The precision, recall, and F1-score of the nominal detector on the 2017 evaluation data.

| | | P | R | F1 |
|---|---|---|---|---|
| English | w/o dict | 73.00 | 54.06 | 62.12 |
| | with dict | 69.91 | 56.55 | **62.52** |
| Spanish | w/o dict | 77.16 | 47.62 | 58.90 |
| | with dict | 75.13 | 53.00 | **62.16** |
| Chinese | w/o dict | 60.41 | 17.42 | 27.04 |
| | with dict | 49.41 | 26.58 | **34.56** |

Table 3: The precision, recall, and F1-score with/without the post-processing dictionaries of the nominal mention detection (top N = 25, tuned from TAC 2016 annotated data by cross-validation). The numbers are for the nominal mention span matches on TAC 2017 evaluation data.

For Chinese and Spanish, the nominals are translated into English by Google translation first. This algorithm is useful for **GPE**s. Note that we get the fine-grained types from the Freebase Mid because we can eliminate some noisy types if we have already disambiguated the named entity.

3. **Surface-MID Majority:** Some nominal mentions, such as "space" (the universe) or "world" (Earth), are usually linked to fixed knowledge base entries regardless of context. Here we introduce a simple majority vote: for each kind of nominal surface, we calculate the surface to MIDs counts. When testing, given the surface we only link the nominal to the majority MID. This trick is useful for **LOC**s.

4. **Combination Freebase Search:** Many of the **ORG**s cannot be linked to a named entity by co-reference, but still link to knowledge base entries. For instance, in the "China's government" example above, the named entity "China" should link to the KB entry for the country, but the nominal "government" should link to the KB entry "Chinese Government". We use a combination search to discover the target knowledge base entry. First, we calculate the majority named entity with type **GPE** (NAM-GPE) in a document, then do the following search in Freebase for each candidate nominal (NOM):

   (a) For English, search "[NOM] [NAM-GPE]", "[NAM-GPE] [NOM]", and "[NOM] of [NAM-GPE]"

   (b) For Spanish, search "[NOM] [NAM-GPE]", "[NAM-GPE] [NOM]", and ''[NOM] de [NAM-GPE]"

   (c) For Chinese, search "[NOM][NAM-GPE]" and "[NAM-GPE][NOM]"

5. **Substring matching of Wikification candidates:** Wikification candidates can also represent potential KB entries to link to. For example, the nominal "政府" should be linked to "中华人民共和国政府" , which is also in the Wikification candidates of "中国" .We also find the majority named **GPE** in a document, gather the Wikification candidates of this GPE-NAM, and check whether the candidate contains the candidate nominal in its title. For Chinese, the title also needs to contain all the tokens of the GPE-NAM.

#### 2.3.2 Nominal Linking Performance

Table 4 summarizes the heuristics we use. Note that if we cannot find any co-referred target using the heuristics we list above, then the default option is to link the nominal to NIL. Table 5 shows the performance and the numbers of each type. Here we assume we have perfect NER and perfect nominal detection so that we can measure our linking accuracy without the influence of other components.

Our linking algorithms show their effectiveness compared with the baseline. Surprisingly, we reach over 50% accuracy on **PER** nominals with only the nearest typed-based co-reference, which means that over half **PER** nominals co-referred to the nearest named entity. Moreover, for the **GPE**s, the Freebase typing provides a rich signal for the co-reference heuristics. For the **ORG**s, it's not enough to do the co-reference: we also need to generate

candidates from Wikipedia or do the combination search.

### 2.4 Coherence

To enforce semantic coherence among predictions at a global level, we use Normalized Google Distance (NGD) (Milne and Witten, 2008; Ratinov et al., 2011) as a coherence measure. Given two Wikipedia pages $p_i$ and $p_j$, their NGD is defined as

$$NGD(p_i, p_j) = \frac{\max(\log(f(p_i)), \log(f(p_j))) - \log(f(p_i, p_j))}{\log N - \log(\min f(p_i), f(p_j))} \quad (1)$$

where $f(p_k)$ is the number of inlinks into page $p_k$, $f(p_i, p_j)$ is the number of inlinks common to the pages $p_i$ and $p_j$ and $N$ is the number of pages in Wikipedia. We use a greedy inference procedure where we first sort the mentions in a document based on the margin between the ranker scores of the first two candidates. We then proceed in the sorted order, by fixing the prediction of a mention to the candidate that maximizes the coherence score with the predictions for the mentions before it.

## 3 Event Nugget Detection and Coherence

The Illinois Event Pipeline was developed based on previous works that UI_CCG submitted to prior TAC evaluations (Sammons et al., 2015; Tsai et al., 2016a). We focused on developing the event system for English by leveraging the most abundant sources of training data. We use pipelined supervised classifiers to extract events (identify event spans and types), determine the realis label for each, and make co-reference decisions respectively. In a multi-lingual setting, we rely on a translation system (e.g. Google Translation) to translate the foreign text into English and then directly apply the English event system to get event output. After that, we devise a way to map them back to the original text based on token level alignment.

### 3.1 English Event Pipeline System

**Pipelined Supervised Classifier**

The TAC Event Nugget Task can be divided into three sub-tasks: 1) event nugget detection with types, 2) realis label assignment, and 3) event co-reference. Our implementation uses a pipelined classification approach to first generate event candidates based on Semantic Role Labeling (SRL) signals. Next, it applies an event nugget classifier to classify each event candidate into different types, including "Non-Event". The following stage applies an event realis classifier on valid events to obtain realis labels; finally, we evaluate the semantic similarity between events via a learned linear function and cluster events with greedy inference.

To summarize, in this pipelined supervised approach, we employ three different classifiers:

1. event nugget classifier: a 34-class multi-class classifier (*33 event subtypes and one non-event class*) to detect event nuggets;

2. event realis classifier: a 3-class multi-class classifier (*actual, general, other*);

3. event co-reference similarity function: a binary classifier (*coref, non-coref*).

**Event Candidate Generation**

We use the Illinois SRL (Punyakanok et al., 2008) to pre-process the input text. We treat all verb and noun predicates as event candidates. We have analyzed the SRL predicate coverage on event triggers in a previous work (Peng et al., 2016); the coverage results are shown in Table 2 of that work, and show that SRL predicates provide good coverage of event triggers. Here, we want good recall since we expect the event nugget classifier to filter out most non-trigger predicates. In addition, we pre-process the input text with the Illinois Named Entity (Ratinov and Roth, 2009; Redman et al., 2016) and Illinois Co-reference (Kai-Wei Chang and Roth, 2012) systems.

**Features for Event Nugget Detection**

Both the event nugget and realis classifiers employ the following set of lexical and semantic features.

1. Lexical features: part-of-speech tag and lemma of tokens in a window size of 5 around the candidate token, plus their conjunctions.

2. Seed features: we use 140 seeds for event triggers. We consider whether a candidate token is a seed or not and generate conjunctions of the matched seed and context seeds.

3. Parse Tree features: path from a candidate token to root, number of its right/left siblings and their categories, and paths connecting a candidate token with other seeds or named entities.

| | English | Spanish | Chinese |
|---|---|---|---|
| FAC | Nearest typed-based co-reference + substring of freebase typing | Nearest typed-based co-reference | Nearest typed-based co-reference |
| GPE | Nearest typed-based co-reference + Substring of freebase typing | Nearest typed-based co-reference + Substring of freebase typing | Nearest typed-based co-reference + Substring of freebase typing |
| LOC | Surface-mid Majority | Surface-mid Majority | Surface-mid Majority |
| ORG | Nearest typed-based co-reference + Combination Freebase Search + Substringof Wikification candidates | Nearest typed-based co-reference + Combination Freebase Search | Nearest typed-based co-reference + Combination Freebase Search |
| PER | Nearest typed-based co-reference | Nearest typed-based co-reference | Nearest typed-based co-reference |

Table 4: The summary of the heuristics we use for each language and each coarse type in nominal linking

| | English | Spanish | Chinese |
|---|---|---|---|
| FAC | 77.95 (254) | 79.90 (204) | 87.86 (173) |
| GPE | 64.89 (225) | 67.23 (238) | 56.29 (421) |
| LOC | 72.14 (140) | 65.79 (152) | 68.02 (519) |
| ORG | 43.89 (483) | 32.95 (522) | 43.05 (734) |
| PER | 71.54 (622) | 63.10 (794) | 71.86 (501) |
| Overall | 63.92 | 57.38 | 60.39 |

Table 5: The nominal linking accuracies over every type and every language on the 2017 evaluation data when we assume other components are perfect. The numbers in the parentheses are the number of mentions.

4. NER features: named entities and their types within a window size of 20 around a candidate token.

5. SRL features: whether a candidate token is a predicate and its role, its conjunction with SRL relation names, and the conjunction of the SRL relation name and the NER types in the context.

6. Co-reference features: co-referred entities with the candidate token, and their conjunctions.

7. ESA (Gabrilovich and Markovitch, 2007) features: top 200 ESA concepts.

8. Brown cluster (Brown et al., 1992) features: brown cluster vector of prefix length 4, 6, 10 and 20.

9. WordNet (Miller et al., 1990) features: hypernym, hyponym and entailment words.

**Features for Event Co-reference**

For event co-reference, we train a classifier to

| | Precision | Recall | F1 |
|---|---|---|---|
| Dev Set | | | |
| Span | 61.40 | 55.46 | 58.28 |
| Type | 50.68 | 44.75 | 47.54 |
| Realis | 41.76 | 36.32 | 38.86 |
| Overall | 33.50 | 32.10 | 30.81 |
| Test Set | | | |
| Span | 53.44 | 41.72 | 46.86 |
| Type | 37.46 | 29.24 | 32.85 |
| Realis | 30.30 | 23.65 | 26.57 |
| Overall | 19.80 | 15.46 | 17.36 |

Table 6: **Event nugget detection results on English.** "Span" indicates the performance where we only consider span match. "Type" further represents event type match plus span span; while "Realis" is for realis label match along with span match. We use TAC 2016 data as the development set, and TAC 2017 data as the test set.

model the similarity between each event nugget pair. Features for this classifier are as follows:

1. Nugget Features: all features defined above for event nugget detection applied on two evaluated events and their conjunctions.

2. Argument Features: all features defined above for event nugget detection applied on SRL arguments (A0 and A1) of two evaluated events and their conjunctions.

3. Entity Features: all features defined above for event nugget detection applied on entities extracted directly through entity co-reference and their conjunctions with nugget features.

4. Pair-wise Features: distance, ESA similarities of two events nuggets and number of co-referent entity mentions covered by SRL arguments attached to two event nuggets.

We implement a greedy inference procedure to look at each detected event nugget from left to right. We make co-reference decisions based on the similarity score of the targeted event nugget and its antecedents (also from left to right).

**Learning and Inference Details**
We include several learning and inference details on our implemented event pipeline system here:

1. Choice of Learner: We choose SVM to train all three classifiers. We use L2 loss and tune C on a development set.

2. Output Filtering: During inference, we only keep events of the 18 types that the task guideline requires after we get results from event nugget classifier.

3. Training Data: We use data from both event nugget tracks in TAC 2015 and TAC 2016. In addition, We also subsample the ACE2005 data to align with the label distribution of TAC 2016 data.

4. Post-processing: We apply heuristic rules on the output of the realis labels for event nuggets. If the detected event trigger is the past tense form of a predicate among a pre-determined set, we set the realis label to "actual".

**Empirical Evaluation**
Event nugget detection results are shown in Table 6. We evaluate the system on the TAC 2016 data as the development set, while we report test numbers according to the submission results from TAC 2017. In Table 6, we report results on the overall event nugget detection as well as performance of each component. "Span" indicates the performance where we only consider span match. "Type" further represents where we have detected events with correct types along with the matched span; while "Realis" indicates where the system detected events with correct realis label along with the matched span. We choose standard precision, recall and F1 as the evaluation metrics.

There is a noticeable performance drop from development set to test set. There are potentially multiple reasons behind it: 1) recall on event detection is not satisfactory, which can be caused by filtering of events detected with types outside of the evaluated 18 types; 2) accuracy of the realis classifier is not competitive since we may need to add more task-specific features for this classifier; 3) other unresolved code bugs. These will be further investigated as future work.

We also break down the performance of event nugget detection with respect to each event type, shown in Table 7. We carry out the experiments on the development set and evaluate the performance of *Span, Type, Realis* match the same way as in Table 6. For our pipelined supervised approach, the results are heavily influenced by the training data size with respect to each event type.

End-to-end event detection and co-reference results are shown in Table 8. In the same way as nugget detection results, we also report numbers on the development and test set, respectively. We utilize standard co-reference evaluation metrics: BCUB, CEAFe MUC, BLANC, and the average of these four metrics.

### 3.2 Spanish Event Pipeline System

We first translate the Spanish documents into English with Google Translation. We then run the English event system as explained above. Finally, we map the identified event nugget back to the original Spanish document based on a word translation table constructed from single word translations acquired from the Translation service.

**Event Nugget Mapping From Translated English To Spanish**
Theoretically, if the translation system produces word alignment information, we can directly

| | Span | Type | Realis | Overall |
|---|---|---|---|---|
| conflict.attack | 51.19 | 41.39 | 34.33 | 28.29 |
| conflict.demonstrate | 83.41 | 68.34 | 55.70 | 46.35 |
| contact.broadcast | 36.77 | 30.23 | 24.70 | 20.63 |
| contact.contact | 29.34 | 25.34 | 19.91 | 16.57 |
| contact.correspondence | 15.30 | 14.15 | 11.34 | 8.97 |
| contact.meet | 67.07 | 54.46 | 44.43 | 36.45 |
| justice.arrestJail | 90.93 | 81.09 | 67.31 | 55.29 |
| life.die | 95.05 | 76.64 | 63.38 | 52.33 |
| life.injure | 97.32 | 96.07 | 79.27 | 65.43 |
| manufacture.artifact | 26.13 | 21.40 | 17.62 | 14.46 |
| movement.transportartifact | 10.16 | 9.57 | 7.36 | 6.11 |
| movement.transportperson | 39.39 | 30.55 | 26.21 | 21.26 |
| personnel.elect | 52.64 | 42.35 | 35.45 | 28.88 |
| personnel.endPosition | 74.27 | 61.65 | 49.95 | 41.32 |
| personnel.startPosition | 21.97 | 16.55 | 13.70 | 11.57 |
| transaction.transaction | 7.49 | 6.88 | 5.31 | 4.30 |
| transaction.transferMoney | 28.19 | 22.97 | 18.62 | 15.74 |
| transaction.transferOwnership | 36.00 | 29.29 | 23.79 | 19.64 |

Table 7: **Event nugget detection results breakdown for each event type in English.** We evaluate on the TAC 2016 data and *Span, Type, Realis, Overall* has the same meaning as in Table 6.

| | BCUB | CEAFe | MUC | BLANC | AVG |
|---|---|---|---|---|---|
| Dev Set | 36.86 | 35.67 | 13.43 | 9.77 | 23.93 |
| Test Set | 24.98 | 23.36 | 12.57 | 8.96 | 17.47 |

Table 8: **Event Co-reference results on English.** We use TAC 2016 data as the development set, and TAC 2017 data as the test set. BCUB, CEAFe MUC, BLANC are standard co-reference evaluation metrics, and "AVG" is the average of these four metrics.

| | BCUB | CEAFe | MUC | BLANC | AVG |
|---|---|---|---|---|---|
| Dev Set | 22.06 | 20.81 | 13.52 | 7.37 | 15.94 |
| Test Set | 15.93 | 15.85 | 3.89 | 3.44 | 9.78 |

Table 9: **Event Co-reference results on Spanish.** We use TAC 2016 data as the development set, and TAC 2017 data as the test set. BCUB, CEAFe MUC, BLANC are standard co-reference evaluation metrics, and "AVG" is the average of these four metrics.

| | Precision | Recall | F1 |
|---|---|---|---|
| Dev Set | | | |
| Span | 49.59 | 44.79 | 47.07 |
| Type | 42.57 | 37.59 | 39.93 |
| Realis | 36.54 | 31.78 | 34.00 |
| Overall | 30.59 | 28.13 | 29.31 |
| Test Set | | | |
| Span | 37.40 | 26.62 | 31.10 |
| Type | 27.96 | 19.90 | 23.25 |
| Realis | 21.17 | 15.07 | 17.60 |
| Overall | 15.26 | 10.86 | 12.69 |

Table 10: **Event nugget detection results on Spanish.** "Span" indicates the performance where we only consider span match. "Type" further represents event type match plus span span; while "Realis" is for realis label match along with span match. We use TAC 2016 data as the development set, and TAC 2017 data as the test set.

map event nuggets back to the original document without any ambiguity. However, in our implementation, such word alignment information is not present. We have to devise a way to choose the correct source language token for the detected English event nugget. To achieve this, we build a word level translation table ahead of time. In the case where we cannot find the exact match in the translation table, we choose the token with the least edit distance.

### Empirical Evaluation

Event nugget detection results are shown in Table 10; end-to-end results with co-reference are shown in Table 9. In the same way that we evaluate English event detection and co-reference, we report numbers on the development set (TAC 2016) and test set (TAC 2017), Manual analysis on the sampled event output reveal that the majority of mistakes come from English event nugget detection instead of the mapping back process.

## References


P. Brown, V. Della Pietra, P. deSouza, J. Lai, and R. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Alla Rozovskaya Mark Sammons Kai-Wei Chang, Rajhans Samdani and Dan Roth. 2012. Illinois-coref: The ui system in the conll-2012 shared task. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*.

G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller. 1990. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

David Milne and Ian H. Witten. 2008. Learning to link with Wikipedia. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 509–518.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

V. Punyakanok, D. Roth, and W. Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2).

L. Ratinov and D. Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*.

L. Ratinov, D. Roth, D. Downey, and M. Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Tom Redman, Mark Sammons, and Dan Roth. 2016. Illinois named entity recognizer: Addendum to Ratinov and Roth '09 reporting improved results. Technical report.

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

Chen-Tse Tsai, Stephen Mayhew, Haoruo Peng, Mark Sammons, Bhargav Mangipundi, Pavankumar Reddy, and Dan Roth. 2016a. Illinois ccg entity discovery and linking, event nugget detection and co-reference, and slot filler validation systems for tac 2016.

Chen-Tse Tsai, Stephen Mayhew, and Dan Roth. 2016b. Cross-lingual named entity recognition via wikification. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*.

Chen-Tse Tsai and Dan Roth. 2016. Cross-lingual wikification using multilingual embeddings. In *Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.