TinkerBell: Cross-lingual Cold-Start Knowledge Base Construction

Mohamed Al-Badrashiny[1], Jason Bolton[5], Arun Tejavsi Chaganty[5], Kevin Clark[5], Craig Harman[3], Lifu Huang[4], Matthew Lamm[5], Jinhao Lei[5], Di Lu[4], Xiaoman Pan[4], Ashwin Paranjape[5], Ellie Pavlick[6], Haoruo Peng[7], Peng Qi[5], Pushpendre Rastogi[3], Abigail See[5], Kai Sun[2], Max Thomas[3], Chen-Tse Tsai[7], Hao Wu[6], Boliang Zhang[4], Chris Callison-Burch[6], Claire Cardie[2], Heng Ji[4], Christopher Manning[5], Smaranda Muresan[1], Owen C. Rambow[1], Dan Roth[6], Mark Sammons[7], Benjamin Van Durme[3]

[1] Columbia University
rambow@ccls.columbia.edu

[2] Cornell University
cardie@cs.cornell.edu

[3] Johns Hopkins University
vandurme@cs.jhu.edu

[4] Rensselaer Polytechnic Institute
jih@rpi.edu

[5] Stanford University
manning@stanford.edu

[6] University of Pennsylvania
danroth@seas.upenn.edu

[7] University of Illinois at Urbana-Champaign
mssammon@illinois.edu

Abstract

In this paper we present TinkerBell, a state-of-the-art end-to-end cold-start knowledge base construction system that extracts entity, relation, event and sentiment knowledge from three languages (English, Chinese and Spanish).

1 Introduction

The TinkerBell team has developed the first end-to-end cold-start knowledge base (KB) construction system for three languages (English, Chinese and Spanish), and achieved top performance at TAC-KBP2017 evaluation. The overall system architecture is presented in Figure 1.

Using our existing high-performing techniques as building blocks, we have improved each component by developing a series of novel methods as follows:

- A joint model of name tagging, linking and clustering based on multi-lingual multi-level common space construction.
- Joint transliteration and sub-word alignment for cross-lingual entity linking: Using pairs of wikipedia titles from interlanguage wikipedia links, jointly model subword alignment and transliteration to compare multi-token name mentions across languages.
- Joint inference between entity discovery and linking (EDL) and slot filling (SF): For the first time we tightly integrated EDL results into SF and achieved significant F-score gains on SF for all three languages.
- Event extraction: developed a novel dependency relation based attention mechanism for event argument extraction.
- Sentiment Analysis (BeSt): we used a target-focused method which we augmented with a polarity chooser and trained for the only-entity-target task.
- Cross-lingual coreference resolution: we

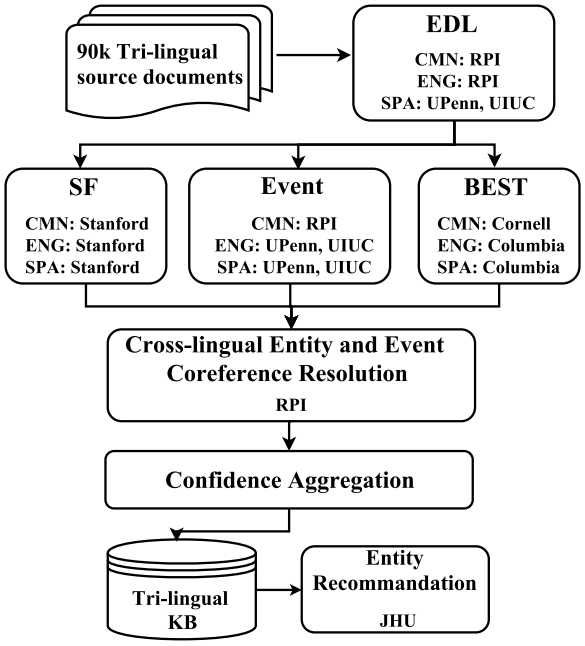

Figure 1: TinkerBell System Overview

developed the first cross-document cross-lingual joint entity and event coreference resolution component.

In the following sections we will present detailed quantitative and qualitative analysis for each component and discuss future research directions. We will also briefly present an entity recommendation demonstration system which searches the tri-lingual KB constructed by TinkerBell and recommend entities for user queries.

2 Entity Discovery and Linking

2.1 English and Chinese EDL

Named Mention Extraction: We consider name tagging as a sequence labeling problem, to tag each token in a sentence as the Beginning (B), Inside (I) or Outside (O) of a name mention with one of five types: Person (PER), Organization (ORG), Geo-political Entity (GPE), Location (LOC) and Facility (FAC). Predicting the tag for each token needs evidence from both of its previous context and future context in the entire sentence. Bi-LSTM networks (Graves et al., 2013; Lample et al., 2016) meet this need by processing each sequence in both directions with two separate hidden layers, which are then fed into the same output layer. Moreover, there are strong classification dependencies among name tags in a sequence. For example, "I-LOC" cannot follow "B-ORG". CRFs model, which is particularly good at jointly modeling tagging decisions, can be built on top of the Bi-LSTM networks. External information like gazetteers, brown clustering, etc. are proved to be beneficial for name tagging. We use an additional Bi-LSTM to consume the external feature embeddings of each token and concatenate both Bi-LSTM encodings of feature embeddings and word embeddings before the output layer. Our model is depicted in Figure 2.

We set the word input dimension to 100, word LSTM hidden layer dimension to 100, character input dimension to 50, character LSTM hidden layer dimension to 25, input dropout rate to 0.5, and use stochastic gradient descent with learning rate 0.01 for optimization.

Nominal and Pronominal Mention Extraction: we utilize a deep neural networks based entity coreference resolution system (Clark and Manning, 2016) in Stanford CoreNLP toolkit (Manning et al., 2014) to extract nominal and pronominal mentions.

Name Translation: We translate Chinese mentions into English based on name translation dictionaries mined from various approaches described in (Ji et al., 2009; Pan et al., 2017). If a Chinese entity mention cannot be translated, we use Pinyin to transliterate it. In addition, we also create a corpus which contains Chinese words and English entities from Chinese Wikipedia, by replacing Chinese anchor links with English entity IDs using cross-lingual links. Using this approach, we learn distributed representations of multi-lingual words and English entities to match Chinese mentions and English candidate entities in the KB.

Entity Linking: Given a set of entity mentions $M = \{m_1, m_2, ..., m_n\}$, we first generate an initial list of candidate entities $E_m = \{e_1, e_2, ..., e_n\}$ for each entity mention m, and then rank them to select the candidate entity with the highest score as the appropriate entity for linking.

We adopt a dictionary-based candidate generation approach (Medelyan and Legg, 2008). In oder to improve the coverage of the dictionary, we also generate a secondary dictionary by normalizing all keys in the primary dictionary using a phonetic algorithm NYSIIS (Taft, 1970). If an entity mention m is not in the primary dictionary, we will use the secondary dictionary to generate candidates.

Then we rank these entity candidates based on three measures: salience, similarity and coherence (Pan et al., 2015).

We utilize Wikipedia anchor links to compute salience based on entity prior:

$$p_{prior}(e) = \frac{A_{*,e}}{A_{*,*}} \quad (1)$$

where $A_{*,e}$ is a set of anchor links that point to entity e, and $A_{*,*}$ is a set of all anchor links in Wikipedia. We define mention to entity probability as

$$p_{mention}(e|m) = \frac{A_{m,e}}{A_{m,*}} \quad (2)$$

where $A_{m,*}$ is a set of anchor links with the same anchor text m, and $A_{m,e}$ is a subset of $A_{m,*}$ which points to entity e.

Then we compute the similarity between mention and any candidate entity. We first utilize entity types of mentions which are extracted from name tagging. For each entity e in the KB, we assign a coarse-grained entity type t (PER, ORG, GPE, LOC, Miscellaneous (MISC)) using a Maximum

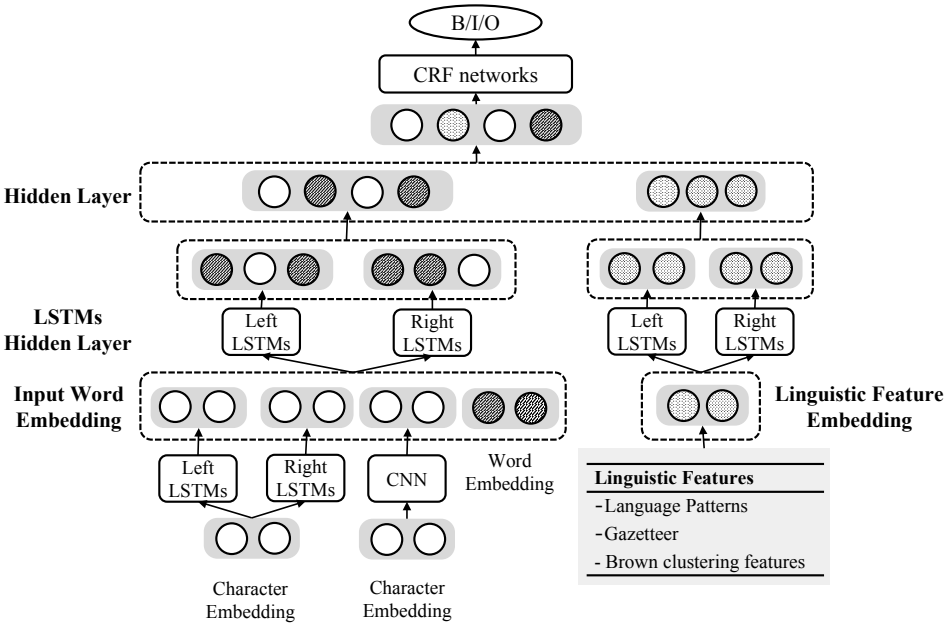

Figure 2: Name Tagging Model with Explicit Linguistic Features.

Entropy based entity classifier (Pan et al., 2017). We incorporate entity types by combining it with mention to entity probability $p_{mention}(e|m)$ (Ling et al., 2015):

$$p_{type}(e|m,t) = \frac{p(e|m)}{\sum_{e \mapsto t} p(e|m)} \quad (3)$$

where $e \mapsto t$ indicates that t is the entity type of e. We also adopt a neural network model that jointly learns distributed representations of words and entities from Wikipedia (Yamada et al., 2017; Cao et al., 2017). Considering all Wikipedia anchor links as entity annotations, a training corpus can be created by replacing anchor links with unique entity IDs. Such training corpus can be used to train the distributed representations of words and entities simultaneously. For each entity mention m, we build the vector representation of its context v_t using the vector representation of each word (exclude entity mention itself and stop words) in the context. Then we compute cosine similarity between the vector representation of each candidate entity v_e and v_t, which can be used to measure similarity between mention and entity $p_{sim}(m,e)$.

Following (Huang et al., 2017), we construct a weighted undirected graph $G = (E, D)$ from DBpedia, where E is a set of all entities in DBpedia and $d_{ij} \in D$ indicates that two entities e_i and e_j share some DBpedia properties. The weight of d_{ij}, w_{ij} is computed as:

$$w_{ij} = \frac{|p_i \cap p_j|}{\max(|p_i|, |p_j|)} \quad (4)$$

where p_i, p_j are the sets of DBpedia properties of e_i and e_j respectively. After constructing the knowledge graph, we apply the graph embedding framework proposed by (Tang et al., 2015) to generate knowledge representations for all entities in the KB. We compute cosine similarity between the vector representations of two entities to model coherence between these two entities $coh(e_i, e_j)$. Given a entity mention m and its candidate entity e, we defined coherence score as:

$$p_{coh}(e) = \frac{1}{|C_m|} \sum_{c \in C_m} coh(e,c) \quad (5)$$

where C_m is the union of entities for coherent mentions of m.

Finally, we combine these measures and compute final score for each candidate entity e.

NIL Clustering: For entity mentions that cannot be linked to the KB, we apply heuristic rules described in Table 1 to cluster these NIL entity mentions. For each cluster, we assign the most frequent entity mention as the document-level canonical mention.

2.2 Spanish EDL

The Spanish Entity Detection and Linking system is based on the Illinois Cross Lingual Wikifier

Rule	Description
Exact match	Create initial clusters based on mention surface form.
Normalization	Normalize surface forms (e.g., remove designators and stop words) and group mentions with the same normalized surface form.
NYSIIS (Taft, 1970)	Obtain soundex NYSIIS representation of each mention and group mentions with the same representation longer than 4 letters.
Edit distance	Cluster two mentions if the edit distance between their normalized surface forms is equal to or smaller than D, where $D = length(mention_1)/8 + 1$.
Translation	Merge two clusters if they include mentions with the same translation.

Table 1: Heuristic Rules for NIL Clustering.

(Tsai and Roth, 2016; Tsai et al., 2016).

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. (2016) 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 Spanish training and evaluation documents, TAC EDL 2016 Spanish evaluation documents, and the Spanish ERE datasets. The model is only trained on the Spanish training data, therefore it is a monolingual model.

Nominal and Pronominal Mention Extraction: For Spanish nominal and pronominal detection, we take Illinois NER as the base model, and only include the following features: the word itself, the neighboring words, and the brown cluster paths of these words. Since other NER features such as gazetteer features and word shape features will not be useful in identifying nominal or pronominal mentions.

We use the nominal mentions in the TAC EDL 2016 Spanish evaluation data as the training examples. We find that including nominal mentions in the ERE dataset does not improve the performance. For pronominal detector, since there is no gold annotation in the previous TAC shared tasks, we only train on the ERE dataset.

Entity Linking: The next step is to ground the extracted Spanish 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 (Mark Sammons, 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.

Co-reference Between Named Entity Mentions and Other Mentions: The above two steps are only performed on the named entity mentions, since the Illinois Cross-Lingual Wikifier focuses on named entities. In the final step, we try to link nominal and pronominal mentions to the named entity mentions, that is, resolving the co-reference problem between nominal/pronominal nouns and named entities.

We apply the following simple heuristic rules to nominal mentions. For each nominal mention, we find the closest mention (either a nominal or a name) to the left which has the same type (PER, ORG, GPE, ...). If this closest mention is a nominal, the surface form is also required to be identical. We then add this nominal mention to the cluster of the closest matching mention. Note that we limit how far do we look ahead by a predefined threshold. If no suitable mention is found within this window, the current nominal mention is discarded, since this nominal mention could refer to some generic noun rather than a specific entity.

We apply different rules for different types of pronominal mentions. The first and second person pronouns usually only appear in discussion forum documents. We try to resolve them to the authors

of posts. We link the first person pronoun to the author mention of the current post. If no author is found, we link it to the previous named entity mention. For a second person pronoun, we link it to the author of the quoted post. If the current post does not quote any previous post (the quoted post will be copied in the current post), we link the second person pronoun to the author of the previous post.

For the third person pronoun, we link it to the previous PER named entity which has the same gender as the pronoun. To determine the gender of a named entity, we count the number of female and male pronouns in the corresponding Wikipedia page. If there are more female pronouns in its Wikipedia page, we classify the entity as female. If no appropriate named entity is found before the target third person pronoun, we simply link it to the previous named entity mention.

3 Slot Filling

Let us now look at the slot filling component of the knowledge base construction system. While the specifics of the slot filling systems for each language (English, Chinese and Spanish) differ, they share the following pipeline: (1) we first construct mention pair candidates for every type-compatible pair of entities in a sentence for each sentence in the corpus (Subsection 3.1), (2) we predict a relation for each of these candidates using one of several relation extractors (Subsections 3.2 – 3.4), and finally (3) we combine the relation predictions from each of the relation classifiers (Subsection 3.5). We will briefly describe the details of each component of the pipeline in this section.

3.1 Mention-pair candidate generation

In the first stage of our pipeline, the entire document corpus is processed using Stanford CoreNLP's annotators (Manning et al., 2014), including a tokenizer, POS tagger, parsers, coreference system and a fine-grained named entity recognition (NER) system. While we are able to use linked entities from the EDL systems (Section 2), the coreference system is critical in extracting relations from pronominal mentions and the fine-grained NER system lets us identify possible slot candidates for the string-valued relations such as `per:title` or `org:date_founded`.

The fine-grained NER system uses a combination of SUTime (English and Chinese) and HeidelTime (Spanish) to identify date expressions and a manually constructed gazette of TokensRegex patterns for each language.

To generate candidates, we simply consider every pair of entity or slot value mentions in a sentence that have compatible types (for example, we might consider a `PERSON` mention and a `TITLE` mention as a valid pair, but not `ORGANIZATION` and `TITLE`).

3.2 Pattern based systems

We have 5 rule-based extractors in total.

Tokensregex and Semgrex. The first set includes a Semgrex pattern system and a TokensRegex (Chang and Manning, 2014) pattern system. TokensRegex patterns search for specific templates (specified via a regular expression) in the word, lemma, POS and NER sequence of a sentence. On the other hand, Semgrex patterns operate on the dependency graph of a sentence and triggers a relation prediction once a specific predefined dependency pattern is matched between two entities.

We reused all patterns from Stanford's 2016 KBP system (Zhang et al., 2016), and added a number of new patterns by hill-climbing on the 2016 development set. Of the two, we found that the dependency based patterns (originally developed in English) were particularly effective at transferring across languages. Of course, it is important to have a high quality dependency parser to use Semgrex patterns: we used the neural depedency parser from Chen and Manning (2014) for English and Chinese and Dozat and Manning (2016) for Spanish. The output of the two pattern extractors are expected to be fairly precise.

Relation-specific extractors. Next we have three relation-specific rule-based extractors: `altnames`, `websites` and `gpe-mentions`. `altnames` is an extractor that infers alternate names of organizations and people from coreference chains of a document, and `websites` compares the edit distance between an organization name and an URL to give high-precision predictions of `org:website` relation. They are described in more detail in Angeli et al. (2015). Finally, `gpe-mentions` uses the occurrence of location names within organization entities (for example `University of [California], [Berkeley]`) to identify location of headquarters relations.

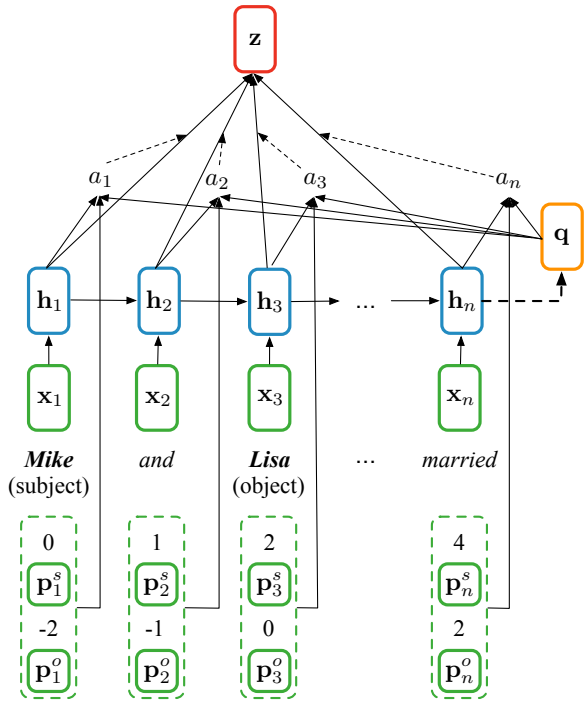

Figure 3: The position-aware neural sequence model for relation extraction. The model is shown with an example sentence *"Mike and Lisa got married."*

3.3 Supervised Logistic Classifier

We reused the self-trained supervised extractor from (Angeli et al., 2015). In summary, at the core of this system is a traditional logistic regression-based classifier with manually-crafted features. We first ran the union of our patterns extractors and an Open IE system on the entire corpus. Since these systems are both of high precision, we collected their positive output predictions to form a training dataset. We added this bootstrapped training set along with a set of presumed negative examples into a pre-collected supervised training set (Angeli et al., 2014), and used this entire dataset to retrain the classifier. We then took this output as the new training dataset, and repeated this process for another iteration. In this way, we trained our statistical models with output from our own classifiers. We also applied other tricks to avoid class skew and overfitting.

3.4 Neural network

Our neural network-based extractor uses an LSTM with position-aware attention (Zhang et al., 2017) as pictured in Figure 3. The model takes the original sentence as input, and generates embedding vectors for each word through a lookup layer which are th fed into an LSTM layer module. We replace the the subject and object entities with special `<subject>` and `<object>` tokens and include features for each token that describe the token offset from the subject and object respectively. When predicting a relation, the output layer attends to a combination of the LSTM outputs and the position features.

Our neural network model is trained on a fully supervised dataset that is constructed from previous years KBP Slotfilling assessment files and is labelled by online crowd sourcing.

3.5 Combining predictions

Once we have relation predictions from each of these systems, we combine their output by simply adding the output probabilities from each system that predicts a relation. Finally, we use a set of heuristic post-processing filters to remove spurious slot fills, e.g. ensuring that `city_of` and `state_of` relations agree.

4 Event Extraction and Coreference Resolution

4.1 English and Spanish Event Extraction and Co-reference Resolution

Since the task includes three sub-tasks (event nugget detection with types, realis label assignment and event co-reference), we use a stage wise classification approach to extract all events.

We first train a 34-class classifier (33 event subtypes and one non-event class) to detect event nuggets and classify them into different types. We then train a realis classifier to decide the realis label for each event nugget. The event type classifier and the realis classifier share the same set of features. During inference, we only keep events of the 18 types that the task guideline requires. For event co-reference, we train a classifier to model the similarity between each event nugget pair. We then 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).

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).[1]. Here, we only focus on recall since we expect the event nugget classifier to filter out most non-trigger predicates. The results show that SRL predicates provide good coverage of event triggers.

Features for Event Nugget Detection
Both the event type and realis classifiers employ the following set of lexical and semantic features.
1) Lexical features: context (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 seed terms for event triggers. We consider whether a candidate token is a seed or not and conjunction 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.
4) NER features: named entities and their types within a window of size 20 around a candidate token.
5) SRL features: whether a candidate token is a predicate or an argument (in which case, its role), its conjunction with SRL relation names and the conjunction of the SRL relation name and the NER types in the context.
6) ESA features: top 200 ESA concepts.
7) Brown cluster features: brown cluster vector of prefix length 4, 6, 10 and 20.
8) WordNet features: hypernym, hyponym and entailment words.

Features for Event Co-reference
For event co-reference, we train a classifier to 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 and their conjunctions with nugget features.
4) Pair-wise Features: distance and ESA similarities of two events nuggets.

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 utilize 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.

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 built ahead of time. Theoretically, if the translation system produces word alignment information, we can directly 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 Spanish 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.

4.2 Chinese Event Extraction and Coreference Resolution

Chinese Event Nugget Detection Event nugget detection remains a challenge due to the difficulty at encoding word semantics and word senses in various contexts. Previous approaches heavily depend on language-specific knowledge. However, compared to English, the resources and tools for Chinese are limited and yield low quality. A more promising approach is to automatically learn

[1]Results are shown in Table 2

effective features from data, without relying on language-specific resources.

We developed a language-independent neural network architecture: Bi-LSTM-CRFs, which can significantly capture meaningful sequential information and jointly model nugget type decisions for event nugget detection. This architecture is similar as the one in (Yu et al., 2016; Feng et al., 2016).

Given a sentence $X = (X_1, X_2, ..., X_n)$ and their corresponding tags $Y = (Y_1, Y_2, ..., Y_n)$, n is the number of units contained in the sequence, we initialize each word with a vector by looking up word embeddings. Specifically, we use the Skip-Gram model to pre-train the word embeddings (Mikolov et al., 2013). Then, the sequence of words in each sentence is taken as input to the Bi-LSTM to get meaningful and contextual features. We feed these features into CRFs and maximize the log-probabilities of all tag predictions of the sequence.

Chinese Event Nugget Realis Prediction For realis prediction, previous methods usually rely on hand-crafted features and dictionaries (Hong et al., 2015). Considering the limited resources for other languages, we apply a Convolutional Nueral Networks (CNNs) (Krizhevsky et al., 2012), incorporating both event nugget and its context information to predict the realis type.

The general architecture of the CNNs is similar as the one we used in our last year's system (Yu et al., 2016). For each event nugget candidate, we apply a standard CNNs framework on both the left and right context of the nugget candidate and concatenate the max-pooling output of both CNNs as well as the representation of nugget candidate as input to a fully connected Multi-Layer Perceptron (MLP). Finally we use Softmax function to predict the realis type.

Chinese Event Argument Extraction For Chinese event argument extraction, given a sentence, we first adopt the Chinese event nugget detection system to identify candidate triggers and utilize the Chinese EDL system to recognize all candidate arguments, including Person, Location, Organization, Geo-Political Entity, Time expression and Money Phrases. For each trigger and each candidate argument, we select two types of sequence information: the surface contexts between trigger word and candidate argument or the shortest dependency path, which is obtained with the Breath-First-Search (BFS) algorithm over the whole dependency parsing output, as input to our neural architecture.

Each word in the sequence will be assigned with a vector, which is concatenated from vectors of word embedding, position and POS tag. In order to better capture the relatedness between words, we also adopt CNNs to generate a vector for each word based on its character sequence. For each dependency relation, we randomly initialize a vector, which holds the same dimensionality with each word.

We encode the following two types of sequence of vectors with CNNs and Bi-LSTMs respectively. For the surface context based sequence, we utilize a general CNNs architecture with Max-Pooling to obtain a vector representation. For the dependency path based sequence, we adopt the Bi-LSTMs with Max-Pooling to get an overall vector representation. Finally we concatenate these two vectors and predict the final argument role with a Softmax function.

Training Details For all the above three components, we utilize all the available Chinese annotation data from DEFT Rich ERE and ACE2005 for training.

Chinese Event Coreference Resolution Our Chinese cross-document event coreference system is composed of two modules: (1) within-document event coreference, and (2) cross-document event coreference.

Within-document Event Coreference: Our within-document Event Nugget Coreference system is based on our last year's system (Yu et al., 2016). We view the event nugget coreference space as an undirected weighted graph in which the nodes represent all the event nuggets and the edge weights indicate the coreference confidence between two event nuggets. And we apply hierarchical clustering to classify event nuggets into event hoppers. To compute the coreference confidence between two events, we train a Maximum Entropy classifier using the features listed in Table 2.

Cross-document Event Coreference: Our cross-document event coreference module is rule based. We merge the hoppers from two documents if they share more than two event arguments.

Features	Remarks(EM1: the first event mention, EM2: the second event mention)
type_subtype_match	1 if the types and subtypes of the event nuggets match
trigger_pair_exact_match	1 if the spellings of triggers in EM1 and EM2 exactly match
stem_of_the_trigger_match†	1 if the stems of triggers in EM1 and EM2 match
similarity_of_the_triggers(wordnet)*	quantized semantic similarity score (0-5) using WordNet resource
similarity_of_the_triggers(word2vec)	quantized semantic similarity score (0-5) using word2vec embedding
POS_match*	1 if two sentences have the same NNPCD
token_dist	how many tokens between triggers of EM1 and EM2 (quantized)
realis_conflict	1 if the realis in EM1 and EM2 exactly match
Entity_match	Number of entities appear both in sentences of EM1 and EM2
Entity_prior	Number of entities appear only in the sentence of EM1
Entity_act	Number of entities appear only in the sentence of EM2

Table 2: Featurs for Classifier. (∗: For English only; †: For English and Spanish only).

5 Belief and Sentiment Extraction

5.1 English and Spanish BeST

Our system is based on Columbia's belief and sentiment system at TAC 2016 (Rambow et al., 2016).

We extended the system for ColdStart++ in the following ways:

- **Data:** We used all the data released prior to the 2016 BeSt eval for training, and the 2016 BeSt eval data for development.
- **Polarity:** Our 2016 BeSt eval system always predicted negative sentiment, as negative sentiment prevails. For ColdStart++ we added a component that identifies sentiment polarity. We chose a system that has high precision on positive sentiment, so that the majority of our predictions remain negative.
- **Confidence:** We added confidence measures which are calculated as a function of the confidence of the classifier and the priors of the target by type.

We discuss polarity in more detail. Table 3 below represents the polarity distribution over the different entities types for entity mentions as targets of sentiment in the English DF data. The table shows that only 4.82% of the entities are the targets of positive sentiment, while 11.46% are the targets of negative sentiment, and 83.72% are not targets of sentiment at all. Due to the very low presence of the positive polarity in the training data, our system tends to favour negative over positive sentiment. To slightly solve this problem, we put the most frequent text among those whose positive polarity in a list. We later use this list to modify the polarity of our output BEST file. So, if the polarity of a certain entity mention is negative and its corresponding text is found in the selected positive words list, we change the polarity to be positive. Table 4 below shows the positive list we are using.

Type	POS	NEG	None
FAC	0.07%	0.21%	2.31%
GPE	0.62%	1.22%	10.51%
LOC	0.08%	0.18%	2.56%
ORG	0.61%	1.71%	8.10%
PER	3.44%	8.15%	60.24%
Total	4.82%	11.46%	83.72%

Table 3: Sentiments polarities distribution over the different types.

al sharpton	israel	navy seals
all of them	it	our troops
all the men in my family	jeb	ows
america	marines	self made man
britain	me	some hard motherf'ers
british	mum	that child
brother	my	the child
country	my family	the country
here	my grandfather	the feds
hes	my mum	the man
i	my parents	the monarchy
iraqis	navy	

Table 4: Positive words list

5.2 Chinese BeST

Our system is based on Cornell's belief and sentiment system at TAC 2016 (Niculae et al., 2016). Specifically, for sentiment, the source extraction component is a rule-based model, which locates source candidates based on post authors and words for reporting speech. The target extraction compo-

#reported	0	1	2	3
$c_{sentiment}$	0.00	0.10	0.30	0.50
#reported	4	5	6	7
$c_{sentiment}$	0.70	0.90	0.95	1.00

Table 5: Chinese BeSt confidence settings. *#reported* refers to the number of system versions reporting the sentiment.

nent consists of (a) a neural network that extracts the polarity of a sentence/mention text/trigger, and (b) a rule-based model that outputs the final polarity of a target candidate based on the output of the model (a) and a bunch of high level features such as target types. Changes made for ColdStart++ are as follows:

- **Data:** More data is used by our system. In particular, the BeSt 2016 eval data is added for model training, and the dictionaries used by our model are extended with more Chinese slangs and idioms.

- **Confidence:** Our system is optimized with F_β measure. Concretely, 7 versions of the system with different β are trained ($\beta^2 = 0.2, 1, 2, 2.5, 5, 10, 50$), and the confidence $c_{sentiment}$ of a sentiment is set according to how many versions by which the sentiment is reported (Table 5). In the submitted runs, we use two different ways to calculate the final confidence c_{final}. One does not take into account entity confidence (i.e. $c_{final} = c_{sentiment}$), while the other does by $c_{final} = c_{sentiment} \cdot c_{target} \cdot c_{source}$.

6 Cross-lingual Coreference Resolution

Cross-lingual Entity Coreference: If two entity mentions share the same entity type and KBID, we consider them coreferential.

Cross-lingual Event Coreference: we only perform cross-lingual event coreference for 'life-die' events. For this specific type of events, we merge two event hoppers in different languages if they share the same victim argument (linking to the same KB entry or having the same NIL cluster ID).

7 Entity Recommendation

Given a set of discovered entities as a query, the Entity Recommendation (ER) task requires a system to find additional entities that are most similar to the queries. We executed this task based on the TinkerBell cross-lingual KB, and here discuss the difference between a content based strategy for ER versus one solely reliant on relations discovered under the TAC relation schema.

For example, TinkerBell discovered the entities *Osama* and *Baghdadi*, and found exactly one entity – *Al Qaeda* – in common within one hop. Using just the extracted relations from TinkerBell, we might then be able to enumerate those other entities that had the same relationship with *Al Qaeda*, e.g. *Suleiman Abu Ghaith*, *Abu Anas al Libi*, *Belmokhtar*, etc. Owing to our prior investigations into the relative sparsity of discovered relations under the TAC schema (with its focus on a small number of high utility relations), our approach relies exclusively on the entity detection and cross-document linking capabilities of the KBP system, then builds models of similiarty based on the content surrounding each of the entity mentions. This allows us in this example to discover entities associated with *Osama* and *Baghdadi* that did not have an explicit relation with *Al Qaeda* in the KB, such as *Ahmed Khalfan Ghailani*, owing to the snippet: *"ghailani had been charged with conspiring in al qaidas 1998 bombings of two u s embassies in east africa."*[2].

Our ER framework currently is supported by two algorithms: "Bayesian Sets" owing to Ghahramani and Heller (2005), and a novel algorithm based on the deep Variational Auto-Encoder framework applied to documents (Miao et al., 2016).

We also provide insight into the working of the entity recommendation system by displaying the most important token features that were used by the recommendation system while scoring the entities.[3] We also include salient mentions associated to each entity which justify why an entity was relevant in the context of a query. Such justifications can aid an analyst in quickly finding supporting data and gaining insights into the text corpus.

Acknowledgments

This work was supported by the DARPA DEFT No. FA8750-13-2-{0041,0017,0040} The views and conclusions contained in this document are

[2] NYT_ENG_20131025.0265

[3] Although our neural variational system does not directly use these features in a linear scoring method, the features are still indicative of the latent space that was constructed and thereby useful in performing error analysis.

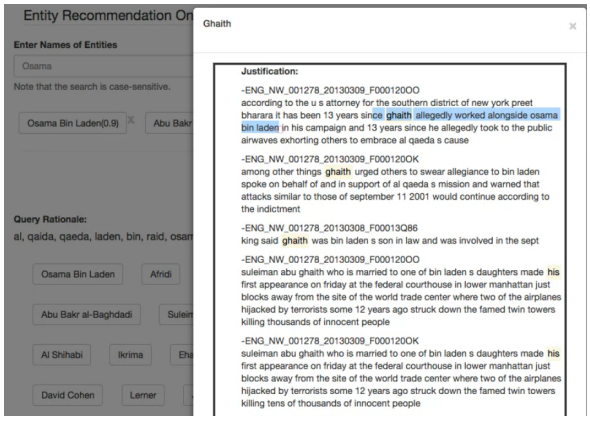

Figure 4: Screenshot of ER system.

those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Gabor Angeli, Julie Tibshirani, Jean Y. Wu, and Christopher D. Manning. 2014. Combining distant and partial supervision for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*.

Gabor Angeli, Victor Zhong, Danqi Chen, Arun Chaganty, Jason Bolton, Melvin Johnson Premkumar, Panupong Pasupat, Sonal Gupta, and Christopher D Manning. 2015. Bootstrapped self training for knowledge base population. In *Text Analysis Conference (TAC) Proceedings 2015*.

Yixin Cao, Lifu Huang, Heng Ji, Xu Chen, and Juanzi Li. 2017. Bridge text and knowledge by learning multi-prototype entity mention embedding. In *Proc. the 55th Annual Meeting of the Association for Computational Linguistics (ACL2017)*.

Angel X Chang and Christopher D Manning. 2014. Tokensregex: Defining cascaded regular expressions over tokens. *Tech. Rep. CSTR 2014-02* .

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 740–750.

Kevin Clark and Christopher D. Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Empirical Methods on Natural Language Processing*. https://nlp.stanford.edu/pubs/clark2016deep.pdf.

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. *CoRR* abs/1611.01734. http://arxiv.org/abs/1611.01734.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Bing Qin, Heng Ji, and Ting Liu. 2016. A language-independent neural network for event detection. In *The 54th Annual Meeting of the Association for Computational Linguistics*. page 66.

Zoubin Ghahramani and Katherine A. Heller. 2005. Bayesian sets. In *NIPS*.

Alan Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding, 2013 IEEE Workshop on*.

Yu Hong, Di Lu, Dian Yu, Xiaoman Pan, Xiaobin Wang, Yadong Chen, Lifu Huang, and Heng Ji. 2015. Rpi blender tac-kbp2015 system description. In *Proc. Text Analysis Conference (TAC2015)*.

Lifu Huang, Jonathan May, Xiaoman Pan, Heng Ji, Xiang Ren, Jiawei Han, Lin Zhao, and James A. Hendler. 2017. Liberal entity extraction: Rapid construction of fine-grained entity typing systems. *Big Data* 5. https://doi.org/10.1089/big.2017.0012.

Heng Ji, Ralph Grishman, Dayne Freitag, Matthias Blume, John Wang, Shahram Khadivi, Richard Zens, and Hermann Ney. 2009. Name extraction and translation for distillation. *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation* .

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. pages 1097–1105.

Guillaume Lample, Miguel Ballesteros, Kazuya Kawakami, Sandeep Subramanian, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceeddings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*.

Xiao Ling, Sameer Singh, and Daniel Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics* 3.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics*.

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

O. Medelyan and C. Legg. 2008. Integrating cyc and wikipedia: Folksonomy meets rigorously defined common-sense. In *Proc. AAAI 2008 Workshop on Wikipedia and Artificial Intelligence*.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *International Conference on Machine Learning*. pages 1727–1736.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Vlad Niculae, Kai Sun, Xilun Chen, Yao Cheng, Xinya Du, Esin Durmus, Arzoo Katiyar, and Claire Cardie. 2016. Cornell belief and sentiment system at TAC 2016 .

Xiaoman Pan, Taylor Cassidy, Ulf Hermjakob, Heng Ji, and Kevin Knight. 2015. Unsupervised entity linking with abstract meaning representation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proc. the 55th Annual Meeting of the Association for Computational Linguistics*.

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)*. http://cogcomp.cs.illinois.edu/papers/PengSoRo16.pdf.

V. Punyakanok, D. Roth, and W. Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics* 34(2). http://cogcomp.cs.illinois.edu/papers/PunyakanokRoYi07.pdf.

Owen Rambow, Tao Yu, Axinia Radeva, Sardar Hamidian, Alexander Fabbri, Debanjan Ghosh, Christopher Hidey, Tianrui Peng, Mona Diab, Kathleen McKeown, and Smaranda Muresan. 2016. The Columbia-GWU system at the 2016 TAC KBP BeSt evaluation. In *Proceedings of the 2016 NIST TAC KBP Workshop*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*. http://cogcomp.cs.illinois.edu/papers/RatinovRo09.pdf.

Tom Redman, Mark Sammons, and Dan Roth. 2016. Illinois named entity recognizer: Addendum to ratinov and roth '09 reporting improved results. Technical report. http://cogcomp.cs.illinois.edu/papers/ner-addendum-2016.pdf.

Robert L Taft. 1970. *Name Search Techniques*. New York State Identification and Intelligence System, Albany, New York, US.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW*. International World Wide Web Conferences Steering Committee, pages 1067–1077.

Chen-Tse Tsai, Stephen Mayhew, and Dan Roth. 2016. Cross-lingual named entity recognition via wikification. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*. http://cogcomp.cs.illinois.edu/papers/TsaiMaRo16.pdf.

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)*. http://cogcomp.cs.illinois.edu/papers/TsaiRo16b.pdf.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2017. Learning distributed representations of texts and entities from knowledge base. *CoRR* abs/1705.02494.

Dian Yu, Xiaoman Pan, Boliang Zhang, Lifu Huang, Di Lu, Spencer Whitehead, and Heng Ji. 2016. Rpi blender tac-kbp2016 system description .

Yuhao Zhang, Arun Chaganty, Ashwin Paranjape, Danqi Chen, Jason Bolton, Peng Qi, and Christopher D. Manning. 2016. Stanford at TAC KBP 2016: Sealing pipeline leaks and understanding chinese. In *Text Analysis Conference (TAC) Proceedings 2016*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 35–45.