

# Stanford at TAC KBP 2017: Building a Trilingual Relational Knowledge Graph

Arun Chaganty,\* Ashwin Paranjape,\* Jason Bolton,\* Matthew Lamm,\*  
Jinhao Lei,\*† Abigail See,\* Kevin Clark, Yuhao Zhang, Peng Qi, Christopher D. Manning  
Stanford University; Stanford, CA 94305

{chaganty, ashwinpp, jebolton, mlamm}@stanford.edu  
{leijjh14, abisee, kevclark, yuhaozhang, pengqi, manning}@stanford.edu

## Abstract

We describe Stanford’s entries in the TAC KBP 2017 Cold Start Knowledge Base Population and Slot Filling challenges. Our biggest contribution is an entirely new Spanish entity detection and relation extraction system for the cross-lingual relation extraction tracks. This new Spanish system is a simple system that uses CRF-based entity recognition supplemented by gazettes followed by several ruled-based relation extractors, some using syntactic structure. We make further improvements to our systems for other languages, including improved named entity recognition, a new neural relation extractor, and better support for nested mentions and discussion forum documents. We also experimented with data fusion with entity linking systems from entrants in the TAC KBP Entity Discovery and Linking challenge. Under the official 2017 macro-averaged MAP all hops score measure, Stanford’s 2017 English, Chinese, Spanish and cross-lingual submissions achieved overall scores of 0.202, 0.124, 0.123, and 0.073, respectively. Under the macro-averaged LDC-MEAN all hops  $F_1$  measure used in previous years, the corresponding scores were 0.254, 0.188, 0.186, and 0.117 respectively.

## 1 Introduction

For the TAC KBP 2017 challenge, we worked to provide a system that handles all three lan-

guages of the multilingual challenge, namely English, Chinese and Spanish, and consequently also the cross-lingual track. The Spanish system was developed entirely from scratch, while both the English and Chinese system received incremental improvements guided by error analysis. We also improved the performance of our entity linking systems in each language by training better named entity recognizers and combining with the entities identified by the RPI (for English and Chinese) and UIUC (for Spanish) entity detection and linking (EDL) systems, made available as part of the Tinkerbell collaboration. We describe the details of our contributions in this paper.

Our English KBP system is built on top of Stanford’s 2016 KBP slot filling system (Zhang et al., 2016). Our two main system improvements were: (1) We enhanced our named entity recognition (NER) system by expanding our training data and building a new neural model. (2) We built a new and improved neural relation extraction system. Our final submission system consists of 5 rule-based relation extractors, a self-trained supervised extractor and a supervised neural network extractor.

Our Chinese KBP system was also built on top of our 2016 slot filling system (Zhang et al., 2016). The 2016 challenge data gave us a first opportunity to do a thorough error analysis of our performance, leading to several minor system improvements and a few major ones: (1) We improved our mention detection by expanding our NER training data and expanding gazettes for fine-grained entity types. (2) We introduced support for nested entity mentions. (3) We hill-climbed on the patterns used during relation extraction. (4) We put in place better document handling of the metadata of discussion forum documents. In our final submission we used the improved pattern based system as well as a distant supervision system.

\*The first six authors all made large, roughly equal contributions to the system.

†Tsinghua University; work completed while interning at Stanford University.

Finally, we developed a completely new Spanish KBP system. This system differs from the English and Chinese system in the following ways: (1) A new Spanish NER system trained on an ensemble of data, and a new fine-grained NER system optimized for KBP, and supplemented by the use of HeidelTime for temporal NER. We also added support nested entity mentions. (2) A limited (pronouns only) coreference system. (3) A new relation extraction system built from pattern-based extractors. We use both dependency tree and token sequence patterns, partially translated from English and then developed manually. (4) The use of a new state-of-the-art neural part-of-speech tagger and dependency parser (Dozat et al., 2017) that were critical to obtain parses of high enough quality to be reliably used by the pattern based system. Our final submission consisted of just the patterns based system.

Additionally, we participated in the cross-lingual track by combining the output of the above three systems. Below, we first introduce the overall pipeline and infrastructure of our KBP systems in Section 2. We then describe improvements to our entity detection and linking system in Section 3. Next, we provide detailed descriptions of our multi-lingual systems: English in Section 4, Chinese in Section 5, and Spanish in Section 6. We present the official evaluation scores of our submissions in Section 7.

## 2 System Architecture

The architecture of Stanford’s 2017 KBP system is largely the same as Stanford’s 2016 and 2015 systems, and is described in detail in [Angeli et al. \(2015\)](#).

In earlier years, our slot filling systems used pipelines starting with an information retrieval (IR) component, which takes the query entities and returns relevant textual mentions and corresponding sentences from the corpus. Then these returned mentions and sentences were passed into downstream relation extractors for further processing. While this architecture has the advantages of being lightweight and saving a lot of computation in the preprocessing phase, it suffers from some critical drawbacks: (1) A fair amount of recall is lost at the beginning of the pipeline, due to the limitations of the IR system. (2) Probing the corpus becomes difficult, as the majority of the corpus remains unprocessed. (3) In order to re-

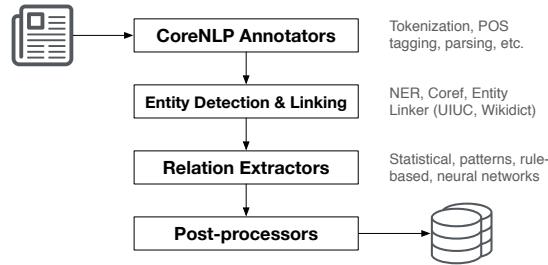


Figure 1: The Stanford KBP system pipeline. The input to the pipeline is a collection of documents, and the output of the pipeline are relation triples stored in database.

trieve multiple candidate mentions in a document, the IR system must be run for multiple iterations.

In this context, we have started to build our KBP system around a relational database, taking inspiration from [Angeli et al. \(2014\)](#). During development, we store all processed documents from the corpus and intermediate data in the database. As most in-database operations are highly optimized and done in-memory, this architecture offers us a lot of benefits. (1) Since the preprocessing component is completely independent from the query entities, we can now annotate all text from the corpus and make use of all potential candidate mentions. (2) Evaluation now becomes optimized database queries instead of full relation extraction cycles, which enables us to do fast iteration on our algorithms. (3) SQL is a powerful data manipulation language that allows us to calculate data statistics and perform system diagnosis quickly.

At the core of this database-centered architecture are two relational tables: **sentence**, which contains textual information about sentences and supporting annotations (e.g., part-of-speech tagging sequences, dependency parsing, etc.), and **mention**, which contains textual information about entity mentions and supporting information of them (e.g., NER tags, provenances, canonical links, etc.). We now describe how the system makes use of this architecture to pipeline different components to produce final output.

### 2.1 System Pipeline

Our full system pipeline is pictured in Figure 1. The input to this pipeline is the original full-text TAC KBP corpus. This corpus is directly fed into an annotation component, where a series of Stanford CoreNLP (Manning et al., 2014) annotators, including tokenizer, part-of-speech (POS) tagger

and parsers, are run to generate structured annotations of the text. The output of this component is used to populate the **sentence** table as described above.

Subsequently, we run our named entity recognition (NER) and coreference resolution annotators to generate NER tags for each sentence and coreference graphs for each document. Then tokens with NER tags of interest are organized together to form entity mentions. For each extracted entity mention, we run an entity linker over it to generate a canonical entity link used to universally describe this entity. We then use the output of this entity detection and linking component to populate the **mention** table, where each mention entry is also connected to its corresponding sentence in the **sentence** table.

As we now have all the annotated information about mentions and their corresponding sentences, we then do simple database join operations on the mention table to form our pool of candidate mention pairs. Note that a candidate mention pair  $(m_1, m_2)$  is generated with the conditions that both  $m_1$  and  $m_2$  are present in the mention table and that they must co-occur in the same sentence in the original corpus. Afterwards, each candidate mention pair is passed into our relation extractors. The output of the relation extractors are a group of scored triples  $(m_1, r, m_2) : p$  where  $r$  is either one of the forward relations as defined in the KBP slot descriptions, or a `no_relation` predicate, and  $p$  is a score that measures how confident the extractor is about this prediction.

Output triples from the relation extractors are then fed into a series of postprocessors. These postprocessors mainly serve three purposes: (1) Inverse relations are generated from all forward relation predictions. (2) Results from different extractors are merged according to our model ensembling policies. (3) We implement some constraints in the postprocessors to filter out predictions that are obviously wrong according to real-world knowledge, and predictions that contradict with others. A more detailed description of this component is presented in [Angeli et al. \(2013\)](#). While the relation extractors are core to the entire system, this postprocessor component is also crucial, as it removes some of the salient errors inevitably generated by the upstream extractors to make sure our system has reasonable precision.

## 2.2 Supporting Infrastructure

We support the pipeline described above with distributed databases, specifically Greenplum DB, set up on two 20-core machines. Doing rapid iterations over the entire pipeline requires intensive large-scale database queries, which greatly benefit from having large memories and fast disk IO speed. Therefore, we set up our machines with 786GB RAM augmented by a 1.2TB PCI-E solid state drive that has a read speed of approximately 2.6GB per second. During development we find this infrastructure setup to be crucial to our quick system testing, problem diagnosis and parameter tuning.

Each language independently uses the same schema and architecture presented above, while the specific implementations of annotators, relation extractors and postprocessors are different across languages.

## 3 Entity detection and linking

In the second stage of our pipeline, we recognize potential entity and slotfill mentions. For each language, there are two systems identify these mentions: a statistical model that predicts entity mentions (i.e. persons, organizations and GPEs) and a rule-based system that identifies fine-grained slot-filling candidates (e.g. titles, religions, etc.). One of the largest sources of error in our system was low recall in our entity detection and linking systems that led to a cascading error in relation predictions. In this section, we address this problem through improvements to our NER systems across languages and a data fusion pipeline to integrate entity predictions from systems that participated in the TAC Entity Detection and Linking track.

### 3.1 Improving NER with Targeted Dataset Expansion

We observed that the documents our NER systems were trained on was very different from the documents found in the KBP corpora, particularly when it came to discussion forum text. Consequently, we augmented the training data for our systems using the DEFT Light/Rich ERE and previous TAC KBP EDL data. Table 1 summarizes the new data used to train our systems and the significant improvements in NER performance that resulted.

Language	New datasets added	Original F1	New F1
English	DEFT ERE Chinese and English Parallel Annotation Data 2014, DEFT ERE English Discussion Forum annotation 2014, TAC KBP EDL Comprehensive Training Data (2014 and 2015), DEFT Rich ERE English Training Annotation Data (2015 and 2016)	75.51	79.99
Chinese	ACE 2004 Multilingual Training Corpus, DEFT ERE Chinese and English Parallel Annotation Data (2014 and 2015), DEFT ERE Chinese discussion forum annotation Data 2014, DEFT Rich ERE Chinese Training Annotation Data 2015, TAC KBP EDL Comprehensive Training Data 2015	66.62	75.90
Spanish	CoNLL 2003 shared task, ACE 2007 Multilingual Training Corpus, DEFT Rich ERE Spanish Annotation 2015, DEFT Spanish Light ERE Training Data 2015, TAC KBP EDL Comprehensive Training Data 2015	54.99	73.18

Table 1: A comparison of NER performance on the TAC KBP EDL evaluation data (2015–2016): we found that augmenting our training data with data from the TAC EDL and DEFT ERE datasets was essential in improving entity recognition performance in all three languages.

### 3.2 Entity Detection and Linking Data Fusion

We observed that the top entrants in the TAC-KBP Entity Detection and Linking track were producing much better quality linking than our systems and we wanted to take advantage of their developments, as our primary objective in the TAC KBP task is developing better slot filling systems. However, the entities predicted by external systems in the EDL track primarily focus on named entities and often do not cover the pronominal mentions and fine-grained slot-value candidates that are essential for good slot filling and are produced by our internal system. To properly resolve pronominal mentions, first consider that we can use the coreference chain to identify a cluster of entities with a unique canonical mention. For every canonical mention identified by our internal system that also appears in the external system’s output, we use the link predicted by the external system for every dependent mention in the coreference chain. There may also be canonical mentions predicted by our internal system that were not predicted by the external system; for these, we use a exact-string-match based heuristic to merge the internal entity cluster with one of the external clusters based on the linked entities name. If this fails, we simply create a new entity id based on the canonical mention’s gloss. Slot-value candidates do not need to be linked and hence we directly use our internal systems’ predictions.

Language	EDL systems	Prec.	Rec.	F <sub>1</sub>
English	Stanford only	55.72	9.61	16.39
	+ RPI EDL	49.81	11.32	18.45
Chinese	Stanford only	27.91	22.64	25.00
	+ RPI EDL	16.50	27.25	20.56
Spanish	Stanford only	28.26	2.49	4.58
	+ UIUC EDL	19.78	3.45	5.87

Table 2: Slotfilling performance observed after integrating EDL predictions from external systems. We find that our recall is significantly improved. The decreases in precision are most likely due to the incompleteness of the evaluation data.

We used the RPI entity linking system (Pan et al., 2017) for English and Chinese, and the UIUC entity linking system (Tsai and Roth, 2016) for Spanish. Table 2 summarizes the improvements obtained through this EDL fusion method. We observed that both significantly improve slotfilling recall, but also significantly decrease precision. A manual inspection in the development set identified that most of the predicted relations are in fact correct and the low precision scores are probably because the evaluation data is highly *incomplete*, as was observed in (Chaganty et al., 2017).

### 4 English KBP System

Our English KBP system refines our 2016 system in several aspects. We summarize the implementation of this system and highlight some of the key

improvements that we made.

## 4.1 Data

We used the TAC KBP 2015 and 2016 slot filling evaluation queries and assessment files to validate and test our English system.

## 4.2 Neural NER Model

In addition to the improvements in entity recognition we proposed in Section 3, we also trained a new neural entity linking model for English. The model uses the bi-directional LSTM-CNN model with CRF decoding proposed in Ma and Hovy (2016). This model led to a 2% increase in F1 scores on the test set described in Section 3, from 79.99% to 82.67%.

## 4.3 Relation Extractors

In total, we use 7 relation extractors that can be broadly divided into three categories.

**Rule-based** We have 5 rule-based extractors in total. The first set includes a *Semgrep* pattern system (Chambers et al., 2007) and a *TokensRegex* (Chang and Manning, 2014) pattern system. *Semgrep* patterns operate on the dependency graph of a sentence and trigger a relation prediction once a specific pre-defined dependency pattern is matched between two entities. In contrast, *TokensRegex* patterns search linearly for specific templates in the word, lemma, POS and NER sequence of a sentence. We reuse all patterns in our 2016 KBP system. The output of the two pattern extractors is expected to be fairly precise.

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 the `org:website` relation. They are described in more detail in Angeli et al. (2015). The *gpe-mentions* extractor identifies nested GPE mentions found in organization entities (e.g., “University of **California**, Berkeley”) to predict `org:<location>_of_headquarters` relations. Further details can be found in Zhang et al. (2016).

**Self-trained Supervised** We reuse the same self-trained supervised extractor as in (Angeli

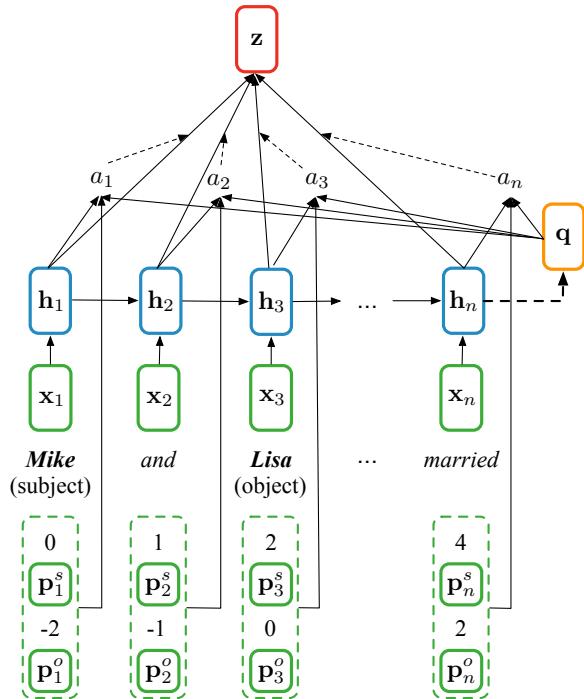


Figure 2: The position-aware neural sequence model for relation extraction. The model is shown with an example sentence “*Mike and Lisa got married*.”

et al., 2015). In summary, at the core of this system is a traditional logistic regression-based classifier (LR) with manually-crafted features and a Long Short Term Memory network (LSTM) classifier. We first run the union of our patterns extractors and an Open IE system on the entire corpus. Since these systems are both of high precision, we collect their positive output predictions to form a training dataset. We add this “bootstrapped” training set along with a set of “presumed negative” examples into a pre-collected supervised training set, and use this entire dataset to train the two core classifiers above (LR and LSTM). We then take this output as the new training dataset, and repeat this process for another iteration. In this way, we train our statistical models with output from our own classifiers. We also apply other tricks to avoid class skew and overfitting as described in Angeli et al. (2015).

**Neural Network-based** This year, we developed a new neural network based extractor that uses an LSTM with position-aware attention (Zhang et al., 2017) as pictured in Figure 2. The model takes the original sentence as input, and generates embedding vectors for each word

through a lookup layer which are fed into an LSTM layer module. We replace 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. We plan to make this dataset publicly available soon.

## 5 Chinese KBP System

This was the second year that we had Chinese slot filling and the first in which we had evaluation data for the cold start task. One of the most significant improvements this year was a better entity recognition system, as detailed in Section 3. For our slot filling system, we improved on our 2016 system (Zhang et al., 2016) with a few error-analysis guided improvements that we describe next.

### 5.1 Data

During development we used the TAC KBP 2016 Chinese Cold Start Slot Filling task evaluation queries and assessment results to evaluate our system. This dataset contains 371 Chinese query entities, which we used entirely for development.

### 5.2 Fine-grained entity recognition

In addition to the improvement on named entity recognition and linking described in Section 3, we also improved on our pattern-based entity and slot-value candidate recognizers. These recognizers use *TokensRegex* patterns (Chang and Manning, 2014) to identify common Chinese entities for the following types: PERSON, ORGANIZATION, COUNTRY, CITY, STATE\_OR\_PROVINCE, TITLE, NATIONALITY, IDEOLOGY, RELIGION, CRIMINAL\_CHARGE and CAUSE\_OF\_DEATH. We further refined our patterns this year based on an error analysis on last year's evaluation data.

### 5.3 Relation Extractors

The lack of a clean supervised dataset continues to limit our use of statistical models during the development of Chinese relation extractors. Thus, we

primarily rely on high-precision pattern-based extractors, though we have also implemented a high-recall distantly supervised extractor. All of these extractors were present in our 2016 system; this year we mainly focussed on tuning the patterns to improve performance on the 2016 evaluation set. We present detailed information of each extractor here:

**Patterns** The development of Chinese *TokensRegex* and *Semgrex* patterns is largely the same as in the English system. We manually add patterns to the extractors that boost the dev set scores, while monitoring the test set scores to avoid overfitting. The final extractors contain 266 *TokensRegex* patterns and 805 *Semgrex* patterns respectively.

**Distantly Supervised Extractor** We build our distantly supervised extractor based around the Mintz system as described in Mintz et al. (2009). In a distantly supervised extractor, training instances are generated by applying a knowledge base to a corpus and labelling each sentence that contains co-occurrence of relation pairs in the knowledge base as a positive instance for the corresponding relation type. In our system, we acquire a knowledge base by using a combination of Freebase relation tuples and relation tuples extracted from previous KBP assessment results. We apply deterministic heuristic rules to convert the Freebase relation types to the KBP slot types. Applying this knowledge base to the Chinese KBP corpus gives us about 530K positive examples for 34 out of the 41 slot types. To balance training data across relation types, we apply a hard threshold of 5K to limit the number of training examples used for each relation type. This finally gives us around 80K training examples in total. Note that unlike the standard setup, we do not use any negative training examples in our distantly supervised extractor, as we empirically find that gradually adding negative training examples will slightly increase precision but decreases recall substantially.

**Other Rule-based Extractors** Additionally, we implement an *altnames* extractor and an *org:subsidiaries/org:place\_of\_headquarters* extractor. The Chinese *altnames* extractor performs inference over the coreference graph to extract `per:alternate_names` and `org:alternate_names` relations. Our *org-subsidiaries* extractor is based on a key obser-

vation that Chinese organization names are often structured in a clearly nested way. For example, in the case of the entity “中国作家协会河北分会”, “中国作家协会” is a parent organization of the former and is nested inside the entity. Therefore, the extractor starts with a training phase where it accumulates a gazetteer of possible organizations by going through all extracted organization entities in the corpus. Then during the extraction phase, it examines the surface string of each extracted organization entity, and if a previously seen entity appears as a substring and this substring satisfies a set of lexical constraints (e.g., the suffix falls inside a lexicon), the extractor generates an `org:subsidiaries` relation for the entity pair of this substring and its full string. Similarly, the `org:place_of_headquarters` extractor looks for location of headquarter from name of organization by matching the substring with a gazetteer containing common GPE entities. We optimize these three rule-based extractors to boost recall while preserving the precision.

## 6 Spanish KBP System

This year, we also developed a new Spanish KBP system from scratch. The system shares the same architectural design as the English and Chinese systems, but has its own relation extraction systems. We’ll discuss the data used during development, our entity recognition components and the relation extractors below.

### 6.1 Data

During development we used the TAC KBP 2016 Spanish Cold Start Slot Filling task evaluation queries and assessment results to evaluate our system. This dataset contains 402 query entities, which we used entirely for development.

### 6.2 Fine-grained entity recognition

As mentioned earlier, the entity recognition system in Section 3 is limited to the 3 entity types, i.e. people, organizations and GPEs. We augmented this system with a *TokensRegex*-based system that uses gazettes to identify common Spanish entities for the following types: COUNTRY, CITY, STATE\_OR\_PROVINCE, TITLE, NATIONALITY, IDEOLOGY, RELIGION, CRIMINAL\_CHARGE and CAUSE\_OF\_DEATH. Much of the gazette was translated from the English version of the same. We also also used

HeidelTime ([Strötgen and Gertz, 2013](#)) to identify time expressions in Spanish.

### 6.3 Rule-based coreference

We also used a simple string-matching based system to link named entities within documents, the results of which were then combined with external EDL predictions. We did not handle coreference with pronouns because Spanish often uses dropped pronouns.

### 6.4 Relation Extractors

For Spanish there is a complete lack of supervised data. As a result, we developed a pattern based system using about 2400 *TokensRegex* and 460 *Semgrep* rules. We initially tried to translate our English patterns into Spanish, but ultimately found that most of the patterns had to be completely rewritten from scratch. To get the *Semgrep* patterns to work reliably, we found it essential to use a high-quality dependency parser: we used the state-of-the-art neural parser from [Dozat et al. \(2017\)](#).

## 7 Results

In this section, we report our evaluation results on the official 2017 evaluation set, using both the new macro-averaged LDC-MEAN average-precision (AP) and slot-filling (SF) scores. The systems we submitted to the cold start KB construction track are described in Table 3, and their results are summarized in Table 5 and Table 6. In addition, we also submitted several systems to the cold start slotfilling track, which are described in Table 4 and their results are summarized in Table 7 and Table 6. Overall, our systems did extremely well, ranking among the top systems for each language.

On the KB track, the diagnostic runs submitted for each language used our expected best slot filling system with different entity detection and linking components, while for the SF track the diagnostic runs used our expected best EDL system with different slotfilling configurations. We were surprised to find that combining EDL systems had mixed results, where it significantly improved slot filling performance in Chinese, but hurt performance in English. Furthermore, the augmented training data for our NER system did not result in significant differences in performance for either English or Spanish. On the other hand, our

<b>System</b>	<b>NER model</b>	<b>Entity linker</b>	<b>Slotfilling system</b>
English			
1	Neural model w/ augmented data	S + RPI	
2	Neural model w/ augmented data	S + UIUC	Pattern-based systems,
<b>3</b>	Neural model w/ augmented data	Stanford	feature-based logistic
4	Linear CRF w/ augmented data	Stanford	classifier and
5	Linear CRF w/o augmented data	Stanford	position-aware LSTM
Chinese			
<b>1</b>	Feature-based model w/ augmented data	S + RPI	
2	Linear CRF w/ augmented data	S + UIUC	Pattern-based systems
3	Linear CRF w/o augmented data	Stanford	
Spanish			
<b>1</b>	Linear CRF w/ augmented data	S + UIUC	
<b>2</b>	Linear CRF w/o augmented data	S + UIUC	Pattern-based systems
Cross-lingual			
1	Combination of system 1 from each language		

Table 3: A summary of the submissions to the KBP 2017 cold start KB construction tracks.

<b>English</b>						
<b>S</b>	<b>NER model</b>	<b>Entity linker</b>	Patterns	Logistic R.	LSTM	Multiple just.
1			✓	✓	✓	✓
2	Neural model w/ augmented data	RPI	✓	✓		✓
3			✓			✓
<b>4</b>			✓	✓	✓	
<b>Chinese</b>						
<b>S</b>	<b>NER model</b>	<b>Entity linker</b>	Patterns	Subsidiaries	Distant sup.	Multiple just.
1			✓	✓		✓
2	Linear CRF w/ augmented data	RPI	✓	✓	✓	✓
3			✓			✓
<b>4</b>			✓	✓		
<b>Spanish</b>						
<b>S</b>	<b>NER model</b>	<b>Entity linker</b>	Patterns		Multiple just.	
1	Linear CRF w/ augmented data	UIUC	✓		✓	
2			✓		✓	

Table 4: A summary of the submissions to the KBP 2017 cold start slotfilling tracks.

slotfilling systems performed as expected: in English, our high recall system (1) did significantly better than the our balanced recall system (2) and high precision systems (3). In Chinese, the different systems we proposed did not lead to significantly different performance. Across languages,

we found that using a single justification led to slightly better performance on the average precision metric (we do not consider precision, recall and F1 scores because they only consider a single justification).

For this track, we used different variants of our

S	Hop-0			Hop-1			All		
	P	R	F <sub>1</sub>	P	R	F <sub>1</sub>	P	R	F <sub>1</sub>
English									
1	23.9%	36.5%	26.1%	19.8%	<b>26.0%</b>	<b>20.6%</b>	22.3%	32.4%	23.9%
2	22.8%	33.5%	23.7%	18.6%	23.8%	19.2%	21.2%	29.7%	22.0%
<b>3</b>	<b>26.1%</b>	<b>38.1%</b>	<b>28.5%</b>	<b>20.1%</b>	25.6%	<b>20.6%</b>	<b>23.8%</b>	<b>33.3%</b>	<b>25.4%</b>
4	24.4%	35.0%	26.0%	16.4%	19.9%	16.2%	21.3%	29.1%	22.2%
5	25.4%	35.3%	26.8%	15.0%	18.2%	15.1%	21.4%	28.7%	22.2%
Chinese									
<b>1</b>	<b>26.9%</b>	<b>23.9%</b>	<b>24.4%</b>	<b>23.1%</b>	<b>6.6%</b>	<b>7.6%</b>	<b>19.6%</b>	<b>18.1%</b>	<b>18.0%</b>
2	17.6%	14.2%	15.1%	13.7%	4.4%	3.8%	12.6%	10.7%	11.1%
3	23.9%	20.9%	21.6%	20.7%	4.0%	2.9%	16.3%	14.9%	14.9%
Spanish									
1	23.8%	24.6%	22.9%	10.8%	10.8%	10.6%	19.2%	19.8%	18.6%
2	23.8%	24.6%	22.9%	10.8%	10.8%	10.6%	19.2%	19.8%	18.6%
Cross-lingual									
1	17.8%	18.3%	16.0%	7.6%	7.9%	7.1%	12.9%	13.3%	11.7%

Table 5: Official scores (macro-averaged LDC-MEAN) of submissions (S) to the KBP 2017 cold start KB construction tracks measured using a single justification.

S	Average Precision		
	Hop-0	Hop-1	All
English			
1	32.3%	<b>11.7%</b>	26.7%
2	30.8%	7.7%	24.9%
<b>3</b>	<b>33.4%</b>	10.7%	<b>27.5%</b>
4	30.5%	6.7%	26.2%
5	31.1%	6.6%	26.3%
Chinese			
<b>1</b>	<b>23.1%</b>	<b>3.3%</b>	<b>18.4%</b>
2	13.7%	1.6%	10.2%
3	0.0%	0.0%	0.0%
4	20.7%	0.7%	16.8%
Spanish			
1	23.5%	4.8%	16.3%
2	23.5%	4.8%	16.3%
Cross-lingual			
1	17.1%	3.5%	11.8%

S	Average Precision		
	Hop-0	Hop-1	All
English			
1	27.1%	8.0%	21.6%
2	23.5%	6.9%	19.0%
3	20.0%	5.7%	16.4%
4	27.4%	9.3%	21.9%
Chinese			
1	22.6%	1.9%	17.4%
2	22.4%	1.9%	17.3%
3	22.6%	1.9%	17.4%
4	22.6%	1.9%	17.4%
Spanish			
1	20.4%	0.8%	13.4%
2	20.9%	0.8%	13.8%
Cross-lingual			
1	15.0%	1.9%	9.8%

Table 6: Official scores (macro-averaged LDC-MEAN) of submissions (S) to the KBP 2017 cold start KB construction (left) and slotfilling (right) tracks measured using up to 3 justifications.

S	Hop-0			Hop-1			All		
	P	R	F <sub>1</sub>	P	R	F <sub>1</sub>	P	R	F <sub>1</sub>
English									
1	25.7%	31.8%	25.7%	18.0%	20.6%	17.7%	22.7%	27.5%	22.6%
2	24.0%	26.4%	23.0%	14.6%	14.8%	14.0%	20.3%	21.9%	19.5%
3	24.3%	21.8%	21.0%	13.0%	12.4%	12.3%	19.9%	18.1%	17.6%
4	27.0%	31.0%	26.3%	19.2%	19.3%	18.1%	24.0%	26.4%	23.1%
Chinese									
1	26.6%	23.8%	24.2%	9.4%	9.7%	9.2%	20.5%	18.7%	18.8%
2	26.2%	23.7%	23.9%	9.3%	9.4%	9.0%	20.1%	18.6%	18.5%
3	26.6%	23.8%	24.2%	9.4%	9.7%	9.2%	20.5%	18.7%	18.8%
4	26.6%	23.8%	24.2%	9.4%	9.7%	9.2%	20.5%	18.7%	18.8%
Spanish									
1	20.7%	21.0%	19.4%	2.9%	3.8%	3.2%	14.4%	14.9%	13.7%
2	21.9%	21.5%	20.5%	2.9%	3.8%	3.2%	15.2%	15.2%	14.4%
Cross-lingual									
1	18.1%	16.5%	15.3%	6.0%	6.0%	5.5%	12.3%	11.4%	10.6%

Table 7: Official scores (macro-averaged LDC-MEAN) of submissions (S) to the KBP 2017 cold start slotfilling tracks measured using a single justification.

slotfilling systems in the diagnostic runs.

## 8 Conclusion

In this paper we have presented the design and implementation of Stanford’s TAC KBP 2017 multilingual slot filling and knowledge base population systems. We explored different methods to improve our entity recognition component and found that improvements in our named entity model (using a neural CRF for English) and combining with high quality EDL predictions (in Chinese) led to significant improvements. We were also able to extend our system to Spanish by utilizing advances in Spanish parsing and building our own pattern-based systems.

## Acknowledgments

The Stanford KBP 2017 team would like to acknowledge the many contributions from participants in the development of Stanford’s previous KBP systems, particularly Gabor Angeli.

Stanford University gratefully acknowledges the support of the Defense Advanced Research Projects Agency (DARPA) Deep Exploration and Filtering of Text (DEFT) Program under Air Force Research Laboratory (AFRL) contract no.

FA8750-13-2-0040. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, AFRL, or the US government.

## References

- Gabor Angeli, Arun Chaganty, Angel Chang, Kevin Reschke, Julie Tibshirani, Jean Y Wu, Osbert Bastani, Keith Siliats, and Christopher D Manning. 2013. Stanford’s 2013 KBP system. In *Proceedings of the Sixth Text Analysis Conference (TAC 2013)*.
- Gabor Angeli, Sonal Gupta, Melvin Jose, Christopher D Manning, Christopher Ré, Julie Tibshirani, Jean Y Wu, Sen Wu, and Ce Zhang. 2014. Stanford’s 2014 slot filling systems. In *Proceedings of the Seventh Text Analysis Conference (TAC 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 *Proceedings of the Eighth Text Analysis Conference (TAC2015)*.
- A. Chaganty, A. Paranjape, P. Liang, and C. Manning. 2017. Importance sampling for unbiased on-demand evaluation of knowledge base population.

In *Empirical Methods in Natural Language Processing (EMNLP)*.

Nathanael Chambers, Daniel Cer, Trond Grenager, David Hall, Chloe Kiddon, Bill MacCartney, Marie-Catherine De Marneffe, Daniel Ramage, Eric Yeh, and Christopher D Manning. 2007. Learning alignments and leveraging natural logic. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. pages 165–170.

Angel X. Chang and Christopher D. Manning. 2014. TokensRegex: Defining cascaded regular expressions over tokens. Technical Report CSTR 2014-02, Department of Computer Science, Stanford University.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford’s graph-based neural dependency parser at the conll 2017 shared task. In *Proceedings of the Twenty-First Conference on Computational Natural Language Learning*.

Xuezhe Ma and Eduard Hovy. 2016. [End-to-end sequence labeling via bi-directional lstm-cnns-crf](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1064–1074. <http://www.aclweb.org/anthology/P16-1101>.

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

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

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*.

Jannik Strötgen and Michael Gertz. 2013. [Multi-lingual and cross-domain temporal tagging](#). *Language Resources and Evaluation* 47(2):269–298. <https://doi.org/10.1007/s10579-012-9179-y>.

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>.

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 *Proceedings of the Seventh Text Analysis Conference (TAC 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.