Stanford's 2013 KBP System

Gabor Angeli, Arun Chaganty, Angel Chang, Kevin Reschke, Julie Tibshirani, Jean Y. Wu, Osbert Bastani, Keith Siilats, Christopher D. Manning
Stanford University
Stanford, CA 94305
{angeli,chaganty,angelx,kreschke,jtibs}@stanford.edu
{jeaneis,obastani,siilats,manning}@stanford.edu

Abstract

We describe Stanford's entry in the TAC-KBP 2013 Slotfilling challenge. Our system makes use of a distantly supervised approach, implementing the multi-instance multi-label system of Surdeanu et al. (2012). In addition, Stanford's system significantly improved the information retrieval component of the system, as well as the consistency and inference procedure applied after candidate relations have been extracted. Stanford's 2013 KBP entry achieved an F_1 of 31.36 on the 2013 evaluation data, performing above the median entry (15.32 F_1).

1 Slotfilling System

The Stanford KBP system used the distantly supervised MIML-RE system (Surdeanu et al., 2012) for relation extraction, adding additional consistency and inference components (see Sections 2.3 and 2.4). In addition, Stanford placed a particular focus on improving the core components of the system, including information retrieval (see Section 2.1) and NER (see Section 2.2).

We describe the the high level architecture of our slotfilling system, and the resources made use of in various components of the system.

1.1 Overview

At a high level, our slotfilling system takes as input a query entity e, and produces a set of slot fills, each of which contains a relation (*slot*) r and a slot value v, while making use of a large unlabeled corpus of text. Our system centers around the MIML-RE relation extractor, which given a set of sentences $\{s\}$ provides:

$$P(r \mid e, v, \{s\})$$

Section 3 describes training the model. At test time, we implement an IR and consistency component to provide relevant sentences to the relation extractor, and filter its output. The IR component is given the query entity e and must find a number of values v which may be in some relation with e, as well as a set of sentences for each value $\{s\}$ which contain both e and v:

$$e \rightarrow \{(v, \{s\})\}$$

The system's IR retrieves candidate documents using Lucene; entity mentions are extracted via direct string match or coreference. Candidate slot values are extracted from spans with a valid NER tag.

After relation extraction we implement a consistency component to ensure that the independent predictions of the relation extractor are mutually consistent and are valid given world knowledge. For example, the relation extractor lacks the world knowledge that one can only be born in one country, or that one cannot be born in a city which is not inside their country of birth. A consistency component is executed after the relation extractor is run over every candidate value v. This component is, at a high level, simply a function from a set of relations R to a subset of those relations R' such that $R' \subseteq R$.

Lastly, the pipeline described above makes the assumption that every slot fill for an entity is expressed in a single sentence somewhere in the corpus, which can be retrieved by IR and classified with the relation extractor. However, in many cases longer term inference must be used to infer relations. Furthermore, making use of predicates which are not in the official set of relations can expose useful inference chains. We present preliminary results of ongoing work on inference for KBP, making use of common ReVerb relation chains to propose new slot fills (see Section 2.4).

1.2 Resources Used

The Stanford system did not make use of the internet during the evaluation period. However, the following resources were downloaded and used offline for both training and evaluation to obtain relevant sentences for the relation extractor:

- The KBP 2010 and 2013 source documents, processed with CoreNLP and indexed with Lucene.
- The July 3, 2013 Wikipedia dump, processed with the wp2txt Ruby package[1] and some simple regular expressions, and further processed with CoreNLP and indexed with Lucene.
- Web snippets, as used in previous versions of the Stanford KBP system.

The CoreNLP pipeline included the default annotators, augmented with the RNN parser of **?**). Furthermore, the NER system was augmented with a collection of 74k regular expression rewrite rules capturing named entity types not recognized by the Stanford NER system (e.g., *Nationality*, *Title*), or refining recognized named entity types into more informative categories (e.g., *Location* $\rightarrow$ *Country*). See Section 2.2 for more details.

The consistency component of the system made use of a number of external resources encoding world knowledge, including:

- A gazetteer, with the raw data extracted largely from `http://www.geonames.org/`, but also including city acronym data from `http://www.allacronyms.com/tag/city`, as well as a mapping from countries to nationalities scraped from Wikipedia.
- A list of nicknames for use in approximate name matching.
- The Wikipedia cross-lingual dictionary (Spitkovsky and Chang, 2012) used in approximate name matching.

2 Models and Algorithms

In addition to incorporating MIML-RE, a number of notable other algorithmic improvements have been made, described below.

[1] `https://github.com/yohasebe/wp2txt`

2.1 Information Retrieval

The IR component of the KBP system was improved significantly. Rather than issuing a fixed query – only the entity at test time, and the entity and slot value at training time – we employ a backoff approach issuing multiple queries progressively increasing recall at the expense of precision until the allotted 50 documents are retrieved. This is similar to approaches taken by **?**) and other question answering systems.

We evaluate our IR component in isolation of the remainder of the system by taking gold provenances in the response file from KBP 2010. In this setting, our IR component correctly retrieves 90% of correct documents when both the entity and slot value are known – as is the case during training – and 66% of correct documents when only the entity is known. For comparison, our initial approach retrieved only 75% of correct documents when both the entity and slot value were known, and only 61% of correct documents when only the entity was known.

2.2 RegexNER

The RegexNER system used in previous Stanford KBP submissions was augmented with more entries to improve recall, boosting the number of evaluation entries tagged with the correct named entity type in the 2010 evaluation corpus from 54% to 69%. Many remaining errors identified for the 2010 entities are caused by Stanford NER incorrectly labeling a portion of the slot value. For example *Lothian* in *Western College of Lothian* marked as a *Location*, or *'s* marked without an NER tag in *Columbia University 's National Center for Disaster Preparedness*. Additional errors include failures in identifying causes of death (e.g., *struck by a car*), and failures identifying compound titles (e.g., *meteorology* in *meteorology professor*, or *male* in *male model*).

2.3 Consistency

The validity of slots and their consistency with each other is enforced with a weighted CSP. The constraints fall into one of three broad categories: constraints on a single slot, pairwise constraints between two slots, or global constraints, checking whether a slot can be enabled conditioned on every other slot already enabled. The natural objective is to maximize the sum confidence of the slots returned subjected to these constraints; how-

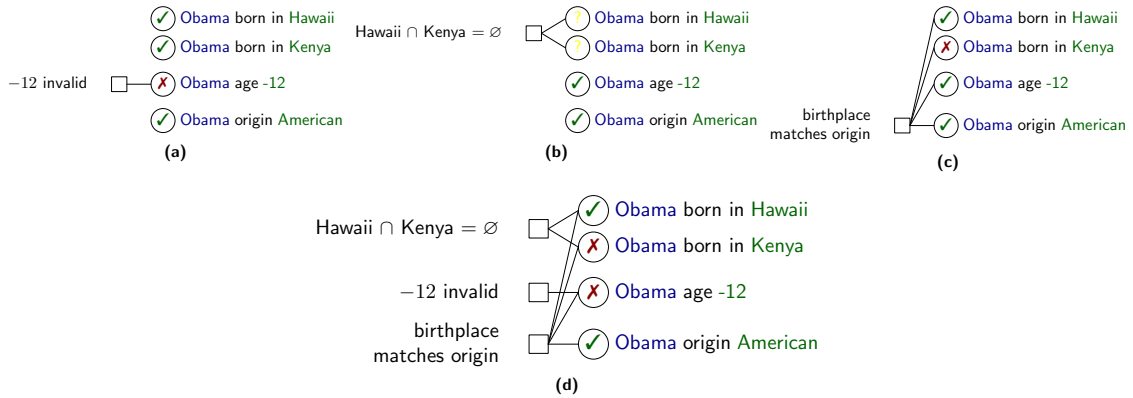

Figure 1: An illustration of the consistency component. The relation extractor proposes four slot fills, shown to the right. Hard constraints are enforced on (a) single nodes, as in the age relation; (b) pairs of nodes, as between the two *born in* relations; and (c) globally, as in the origin relation. (d) shows the CSP with all of the above factors included. See Section 2.3 for a full list of factors.

ever, in practice we get similar accuracy by greedily enabling slots from most to least confident, if they do not violate any constraints over slots already enabled. In addition, a number of deterministic rewrite rules handle cases which are difficult for the relation extractor, usually pertaining to nuances in the evaluation guidelines.

Figure 1 shows an example of the types of factors included in the CSP; the full list of constraints and rewrites are:

- Filter relations which do not match the entity type. For instance, `per:origin` for an organization.
- Filter URLs which do not sufficiently overlap with the name of the entity, and which do not represent an acronym or partial acronym of the entity.
- Filter very long slot values (> 80 characters).
- Filter ages which are not between 0 and 125.
- Filter slot fills which violate the examples given in the annotation guidelines, e.g., invalid titles.
- Filter invalid countries, states, and cities.
- Filter duplicate slots, both exact duplicates and approximate duplicates. Approximate duplicate slot values were calculated from token overlap, and simple rules on the Wikipedia cross-lingual dictionary.
- Filter relations which empirically have never co-occurred in the knowledge base.
- Filter location and date of death unless the other is known as well.
- Filter invalid geographic relationships, such as enforcing that an entity's birth country and birth state are consistent. Furthermore, ensure that an entity has a reasonable number of origins, and prefer that their origin matches their country of birth.
- Rewrite a slot value to its canonical mention as determined from coreference.
- Rewrite titles to their most specific form, to mitigate errors in RegexNER described in Section 2.2.
- Rewrite *born in* and *founded by* relations to *resides in* and *top employee of*, unless there is explicit evidence in the sentence that the born in or founded by relation should hold.

Lastly, a component was developed to enforce that a relation and slot value only hold for a single entity in a sentence – that is, enforces a notion of sentence-level competition between slot fills. For instance, if a sentence has candidate slot value v and entities e and e', then disallow the relation extractor proposing both (e, r, v) and (e', r, v).

2.4 Inference

An inference component was developed for proposing new slots not inferred from the relation extractor. The component builds a graph between entities, with edges corresponding to relations between the entities (not necessarily KBP relations). Two versions of inference were submitted to KBP 2013. A conservative version enforces symmetry between a relation and its inverse, and merges entities based on the similarity of their properties

in addition to the similarity of their surface string form.

In addition, a version was submitted which scrapes common ReVerb patterns between training entity, slot value pairs, and uses this information to propose new slot fills when these patterns occur in the test data. These are preliminary results providing a baseline for more sophisticated inference rules over the graph, both in the collection and weighting of the rules, and the inference at test time from rules collected.

Lastly, geographic relations were proposed where they are clearly entailed. For example, if the city of birth is known for an entity, and the city can be uniquely identified as belonging to a state and country, the slot values for the state and country of birth of that entity are proposed.

2.5 Rule Based Extraction

A number of rules were created manually, constructed primarily based on error analysis over the 2010 corpus. Each rule is expressed as a TokensRegex pattern, which is run over every sentence returned by IR, and proposes a given relation upon matching the pattern. A total of 69 such rules were constructed, over a total of 15 slot types. The slot with the most rules defined was `per:schools_attended`; the list of rules defined for this slot is given in Table 1, and is illustrative of the types of rules defined for the other slots.

3 Training

For training, we used relation tuples extracted from Wikipedia infoboxes. We converted Wikipedia relation labels to KBP slot types using a deterministic in-house process. These tuples were then aligned with text from the three data sources described in Section 1: the official docs from 2010 and 2013, and Wikipedia from July 2013.

Negative examples were collected from IR proposals (e, v) which do not match any known entry in our knowledge base; that is, for which there is no relation that contains v as the slot value. In more detail, we propose a negative example (e, r, v) if it is subsampled from the pool of candidate (e, v) pairs (10% for the 2013 evaluation), and either of the following criteria hold:

1. We know that there exists an entry in our knowledge base (e, r, v') $v \neq v'$, and r is a single-valued relation. That is, we have contradictory evidence in our knowledge base.
2. We know that there exists an entry in our knowledge base (e, r', v) such that r and r' are incompatible.

This ensures that the negatives we add are of relatively high quality; a clear improvement would be adopting the methodology of **?**) to treat examples as unknown initially, and add them as negatives gradually.

MIML-RE was trained for 8 iterations using 3 folds. No model combination was attempted.

4 Results

We report a number of results on the official 2013 evaluation set, as well as various development sets used for improving and tuning our system.

4.1 Official Scores

Stanford submitted 5 systems for the official evaluation. For all runs, a fixed confidence threshold of 0.5 was imposed based on the tuned threshold on the 2012 data. Slot fills under this threshold were discarded, with the exception of inferred slots which were always kept. These are described in Table 2 and in more detail below, in order of expected performance:

S1 The reference run, incorporating every component except inference using ReVerb relations.

S2 S1, but with all inference components described in Section 2.4 disabled.

S3 S1, but with the experimental ReVerb inference paths from Section 2.4 enabled.

S4 S1, but with inference (Section 2.4), sentence-level competition from Section 2.3, and the hand-coded rules from Section 2.5 disabled. This run represents our system run with only MIML-RE and basic consistency.

S5 S1, but using only the 2013 docs for searching for slot fills at test time. Thus, the component of our system which searches for provenance given a slot fill found in another corpus is not relevant.

Previous Rankings Stanford participated in the KBP Slotfilling task between 2009 and 2011. The submitted system placed 5^{th} in 2009, below 5^{th} in 2010, and 4^{th} in 2011.

```
$ENTITY /attended/ /the/? $SLOT_VALUE /[Cc]ollege|[Uu]niversity|[Ss]chool/
$ENTITY /attended/ /the/? /[Cc]ollege|[Uu]niversity|[Ss]chool/ $SLOT_VALUE
$ENTITY /enrolled/ /in/ /the/? $SLOT_VALUE /[Cc]ollege|[Uu]niversity|[Ss]chool/
$ENTITY /enrolled/ /in/ /the/? /[Cc]ollege|[Uu]niversity|[Ss]chool/ $SLOT_VALUE
$ENTITY /enrolled/ []{0,5} /degree|major|program/ /at/ /the/? $SLOT_VALUE /[Cc]ollege|[Uu]niversity|[Ss]chool/
$ENTITY /enrolled/ []{0,5} /degree|major|program/ /at/ /the/? /[Cc]ollege|[Uu]niversity|[Ss]chool/ $SLOT_VALUE
$ENTITY /educated/ /at/ /the/? $SLOT_VALUE /[Cc]ollege|[Uu]niversity|[Ss]chool/
$ENTITY /educated/ /at/ /the/? /[Cc]ollege|[Uu]niversity|[Ss]chool/ $SLOT_VALUE
$ENTITY /graduated/ /from/ /the/? $SLOT_VALUE /[Cc]ollege|[Uu]niversity|[Ss]chool/
$ENTITY /graduated/ /from/ /the/? /[Cc]ollege|[Uu]niversity|[Ss]chool/ $SLOT_VALUE
```

Table 1: Rules employed by the KBP system for the `per:schools_attended` relation. All rules were constructed using TokensRegex; `$ENTITY` and `$SLOT_VALUE` denote the location of the entity and slot value between which the relation should hold; note that other tokens may still also be a part of the entity and slot value.

System	2013 P	2013 R	2013 F_1
S1	35.75	27.93	31.36
S2	35.86	**28.41**	**31.70**
S3	35.06	26.70	30.32
S4	35.31	25.61	29.69
S5	**38.24**	26.70	31.45

Table 2: Stanford's submissions for 2013. The expected best system is S1; S2 removes inference; S3 adds ReVerb entailment rules; S4 removes all inference and rules, relying only on MIML-RE; S5 is identical to S1 but run only over the official 2013 source documents.

System	2013 P	2013 R	2013 F_1
Median Team	14.99	15.67	15.32
Stanford 2013	35.75	27.93	31.36
Top 1 Team	**42.53**	**32.17**	**37.28**

Table 3: Stanford's 2013 KBP results, as compared to Stanford's 2011 submission, and other teams this year.

Year	P	R	F_1	Tuned F_1
2010	27.45	24.82	26.07	27.77
2011	20.14	23.82	21.82	23.47
2012	27.46	19.22	22.61	22.97

Table 4: Development results on the 2010, 2011, and 2012 corpora. For all evaluation, any document was accepted as provenance; the *Tuned* F_1 column reports F_1 if the slot threshold is tuned to its optimal value. The 2010 corpus was used for development; 2012 was used for tuning; 2011 was unused but is included for completeness.

4.2 Development Scores

We report results on various development corpora. All results are reported with `anydoc` set to true. Thus, the results are potentially optimistic, but nonetheless penalized by the incompleteness of the score files, and many slots we propose on these corpora are in fact correct but are marked wrong at evaluation time. Results are reported both without slot thresholding; note that when thresholding is enabled on dev runs, the threshold is tuned to maximize F_1 rather than the fixed 0.5 value for the official runs.

We report results on three corpora: 2010, 2011, and 2012. The 2010 corpus was used for system development and debugging. 2012 was used for final tuning of hyperparameters and determining the expected performance and ordering of submissions. 2011 was not used prior to the evaluation, but is reported here for completeness. The results are reported in Table 4.

In addition, we report results comparing our 2013 entry with other entries this year, and our previous 2011 entry, in Table 3

Results for the 2012 evaluation corpus comparing performance with and without inference enabled is given Table 5, showing the precision and recall without tuning slot thresholds.

5 Slotfilling Error Analysis

We discuss remaining sources of error for our slotfilling system. For this, we take as a starting point a perfect slotfilling system, and gradually replace each of the components with the corresponding

Configuration	2012 P	2012 R	2012 F_1
Inference On	27.46	19.22	22.61
Inference Off	27.22	19.35	22.62

Table 5: Results of the KBP system on the 2012 evaluation corpus, without a slot threshold enabled. *Inference On* denotes our expected best system (S1). *Inference Off* denotes our expected second best system (S2).

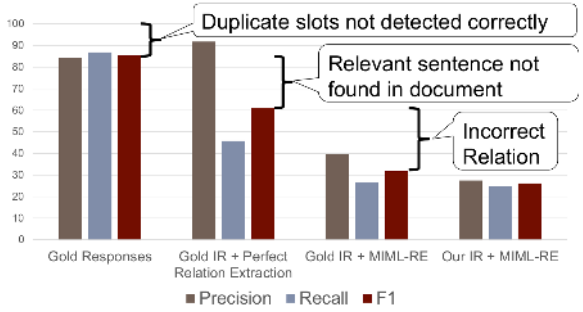

Figure 2: A summary of errors in our KBP system. The experiments take a perfect slotfilling system, and gradually add more components from our system to assess where accuracy drops most significantly. The experiments are described in detail in Section 5.

component from our system. Results are summarized in Figure 2, using the 2010 corpus. The configurations are:

Gold Responses Propose every slot in the evaluation file, passed through the duplicate slot detector, achieving an F_1 of 85.5. This evaluates the effectiveness of the duplicate slot detector; perfect entity linking would fix these errors.

Gold IR + Perfect Relation Extraction Return every and only documents marked as correct provenance for a given entity; and, create a relation extractor which always returns the correct relation if and only if a given entity, slot value pair is proposed to it. Compared to the above system, this system misses slots which either are not expressed in a single sentence and require longer-range inference, or which were not correctly found in the document due to errors earlier in the pipeline – most prominently in coreference or mention detection. This achieves an F_1 of 61.0.

Gold IR + MIML-RE Return every and only documents marked as correct provenance, however run MIML-RE as the relation extractor. Compared to the the above system, errors here are caused by incorrect relation predictions from MIML-RE. This achieves an F_1 of 31.9 (compared to 27.8 for our full system).

The conclusions we draw from these experiments are reflected in the future work we intend to pursue. The errors from incorrectly deduplicating entries would be helped by incorporating an entity linking system. The second class of errors – from not finding a sentence which adequately expresses the target relation – we intend to address by improving our inference component to collect better weights for inferential paths, and to perform more holistic inference on the entity graph at test time with Markov Logic. The third class of errors – incorrect relation predictions – we hope to mitigate by collecting crowd-sourced labels for the latent variables in MIML-RE. In part, this would provide valuable high-quality supervised training data, and in part it is likely to make the model's objective more convex and manageable.

We believe that the relatively small loss incurred from using our IR versus the Gold IR implies that our IR system performs well enough that it is not a bottleneck in improving performance on the task.

6 Conclusion

We have presented Stanford's entries in the 2013 TAC-KBP tasks. Our most significant improvements over previous entries are a new IR system, making use of increasingly general queries to produce better results; incorporating MIML-RE as the relation extractor; and creating a consistency module which enforces both local and non-local constraints to produce consistent slot predictions. We obtained an F_1 of 31.36, performing well above the median team. We entered the consistency component of our system into the slot validation task.

References

Valentin I. Spitkovsky and Angel X. Chang. 2012. A cross-lingual dictionary for English Wikipedia concepts. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, Istanbul, Turkey, May.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *EMNLP*.