Extracting Multilingual Relations under Limited Resources: TAC 2016 Cold-Start KB construction and Slot-Filling using Compositional Universal Schema

Haw-Shiuan Chang **Abdurrahman Munir** **Ao Liu**
Johnny Tian-Zheng Wei **Aaron Traylor** **Ajay Nagesh**
Nicholas Monath **Patrick Verga** **Emma Strubell** **Andrew McCallum**
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA, 01003, USA
hschang@cs.umass.edu

Abstract

We describe the UMass_IESL relation extraction system for TAC KBP 2016. One of the main challenges in TAC 2016 is to extract relations from multiple languages, including those with relatively low resources like Spanish. To mitigate the problem, we integrate multilingual and compositional universal schema from Verga et al. (2016) into our slot filling and knowledge base construction pipelines. The flexibility of our universal schema framework allows us to extract high quality Spanish relations based on English training data, and easily incorporate various types of data collected from Internet. Finally, we show how the improvements of each component contributes to the final scores of our submissions.

1 Introduction

Language resources play an important role in the performance of relation extraction systems. Collecting more resources have been shown to be an effective way to improve the quality of the slot filler and the constructed knowledge base (Angeli et al., 2014). However, the high cost of human annotations impairs the scalability of the approaches. This motivates us to explore ways to further exploit existing resources or collect more online resources without involving an expensive human annotation. In the TAC KBP 2016 Cold Start track, we utilize more online structured data such as Wikipedia redirect pages, transfer English resources to perform better relation extraction in Spanish, and use a search engine to reduce the noise in the distant supervision data. The evaluation results of previous years and this year demonstrate the effectiveness of these approaches.

2 System description

As shown in Figure 1, the pipelines are roughly the same as our 2015 systems (Roth et al., 2015). That is, for the Knowledge Base (KB) construction task, we perform NER (Named Entity Recognition), entity linking, and classification of the textual patterns between entity pairs within the same sentence. For each hop of the Slot Filling (SF) task, we search the expanded queries in the corpus, use NER to filter out the answer candidates with the wrong types, and use the same classifier models to compute the confidence of the answers. In the following

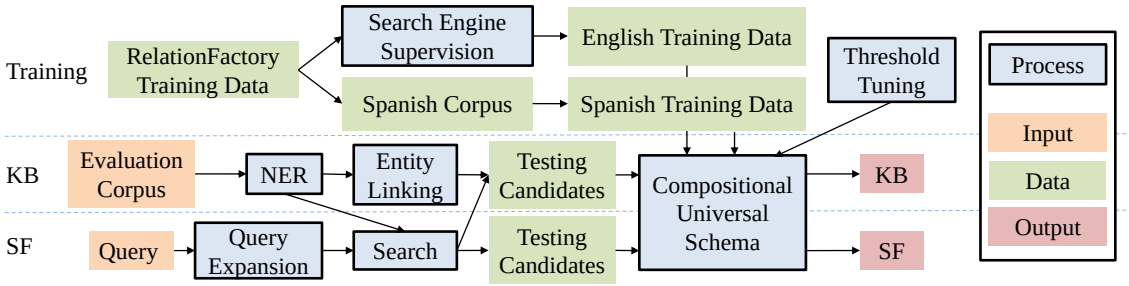

Figure 1: The workflow for our 2016 TAC KBP system. The system solves the Slot Filling (SF) and Knowledge Base (KB) construction using compositional universal schema. The legend on the right shows the color code of each block.

sections, we briefly describe how each component works at test time, and highlight the improvements we made this year. For the sentence classifier module, we also introduce the methods of training and tuning the models.

2.1 Entity Detection and Linking

2.1.1 Named Entity Recognizer (NER)

We use CRF-based NER (Passos et al., 2014) in FACTORIE[1] (McCallum et al., 2009) to detect entities and identify their types. In this year's competition, we focus on improving our Gazetteer features, a list of lexicons with a specific type of interest. Though Gazetteer is often useful when we encounter ambiguous document contexts, the list often contains the surface form of some entities which rarely appear as the type of interest. For example, "ocean" could be used as a person's name, but most of the time it means "sea". Such noise might force the NER tagger to lower the feature weights and effectiveness of such language resources.

To mitigate the problem, we apply a simple entity linker to the lexicon lists. Specifically, we filter out the lexicons which appear 100 times as Wikipedia anchor text, but links to an incompatible Freebase (Bollacker et al., 2008) type at most occurrences. After cleaning Spanish lexicon lists, the F1 error reduces 13% around in CoNLL 2002 (F1: $0.844 \rightarrow 0.865$). For English, the same method reduces the 9% of recall errors for the entity in the TAC 2015 SF queries.

In the SF task, we are not required to output a fixed type for each entity. To increase the recall of some of our SF runs, we link each entity extracted by NER to Freebase through Wikipedia anchor text. If the type of linked Freebase entity is different from the one found by NER, we allow the entities to have types from both NER and Freebase at the same time.

2.1.2 Entity Linking

The entity linking component uses embedded representations of mention contexts to link entity mentions to Wikipedia entities. Three separate representations of contexts and entities are learned; these are an embedding of the local word context, of the co-occurring mentions, and of the co-occurring entities. The embedding model predicts an entity vector based on the sum of context embedding (i.e., a continuous bag of words model Mikolov et al. (2013)).

The embedding are trained on Wikipedia[2],

[1] http://factorie.cs.umass.edu/

[2] We used the dump from 20160305.

with anchor text links as linking annotation. Separate embedding models are learned for English and Spanish. At test time, the linking procedure uses a greedy method. Each entity mention is scored. The entity mention with the highest similarity score is linked and the procedure repeats for the remaining mentions with the entity context updated with the latest link. Unlinked mentions are clustered a simple entity type and surface form matching method. Finally, we split entity clusters based on the NER type associated with the mentions.

The scoring procedure measures the similarity between the mention context and the entity representations for a set of candidate entities determined by alias information derived from Wikipedia anchor texts. The score combines the prior probability of the mention referring to a candidate entity as determined by the Wikipedia anchor texts and features extracted from the distributed context representations (cosine similarity between context and entity; entity-entity coherence measures). Several recent works explore similar models for entity linking (Sun et al., 2015), (Yamada et al., 2016), (Francis-Landau et al., 2016), (Nguyen et al., 2016), (Huang et al., 2015).

In the multiple language setting, Spanish Wikipedia titles are mapped to English titles using the interlanguage page links provided in the Wikipedia dump download[3]. This allows us to answer English query from Spanish corpus, and vice versa.

Our new NER and embedding linker result in a better score in TAC EDL 2016 when we consider strong typed mention match metric. For linking only English entities, The precision improves from 0.723 to 0.818, the recall improves from 0.138 to 0.141, and the F1 improves from 0.231 to 0.240 compared with the linker we used in our 2015 system. Notice that though being encouraged in the TAC 2016, we choose to neglect the nominal and nested mentions because of the limitations of our current coreference component. Furthermore, we ignore facility (FAC) and location (LOC) types for English, due to the limitation of our current training data.

2.1.3 Query Expansion

Previously, we relied mainly on Wikipedia anchor text statistics to find the alias of the entities in the query. In our current system, we find that Wikipedia redirect pages can provide us more aliases of each entity, including additional alternate names, misspellings, etc. For example, searching "Obama" in Wikipedia would be directed to the "Barack Obama" page, so we know that "Obama" is an alias of "Barack Obama". The Wikipedia redirect pages contain some noise. We filter out all the redirect pages with page view less than 10% of all the redirect page view for a specific entity. The further expansion increases the F1 of TAC 2015 SF hop0 queries from around 0.21 to around 0.23. For the cross-lingual setting, we find the alias in another language by relying on the mapping between English Wikipedia titles and Spanish ones as we did in the entity linking.

After we combine the alias from anchor text and from redirect pages, each query is further expanded using some simple manual rules. For example, "Barack Obama" is a person's name, so we also add "B. Obama" into our alias list. Then, we search the relevant documents using Lucene[4] based on the expanded queries. Now that the documents have high probability to be related to the query, we expand the query even further using more manual rules in this year to search the candidates of relevant sentences. For

[3] enwiki-20160305-langlinks.sql.gz

[4] https://lucene.apache.org/

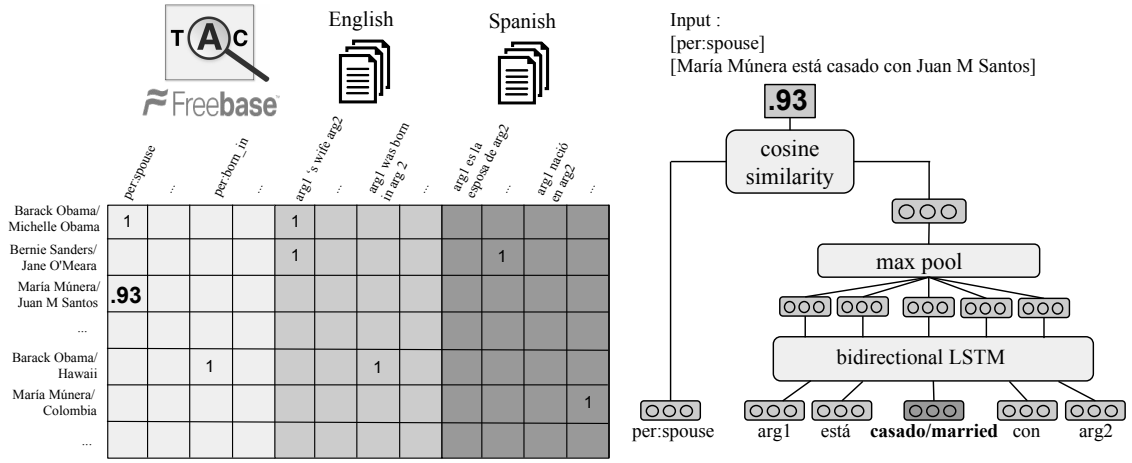

Figure 2: The compositional universal schema model from Verga et al. (2016) learns the embedding of the entity pair on each row and the relation between them on each column. After observing the co-occurrence of entity pairs and relations (i.e., the 1 in the cells of the left matrix), we can train the LSTM on the right to learn the mapping between the token sequence and the embedding of relations. Given a new English or Spanish sentence at testing time, the classification confidence is computed by the cosine similarity between the embedding of the new sentence and the embedding of TAC schema.

example, we would allow the mentions of only last name or only first name of a person to be an alias of the entity in the query. Such within document expansions can improve the F1 of 2015 TAC SF hop0 queries from around 0.24 to around 0.25.

2.2 Compositional Universal Schema

Universal schema is a relation extraction approach based on matrix completion (Riedel et al., 2013). In TAC KBP, we use universal schema as a sentence classifier. As shown in Verga et al. (2016), integrating a Long Short-Term Memory (LSTM) model into the universal schema framework can greatly improve the recall, and the approach is also called compositional universal schema. The recent study from Adel et al. (2016) also demonstrates that neural network can outperform traditional model like SVM. Motivated by the success, we replace the SVM sentence classifier from RelationFactory (Roth et al., 2014) with the LSTM model.

During testing time, the entity linking or query expansion would output many candidates of entity pairs within the same sentence. After extracting the textual patterns between the entity pairs, we run the LSTM to compute the embedding for compositional universal schema, and perform a table lookup to find the embedding for universal schema. As shown in Figure 2, the cosine similarity between the embedding of the sentence and the embedding of each relation in TAC schema would be the confidence of such relation existing between the entity pair. After we apply thresholds to the

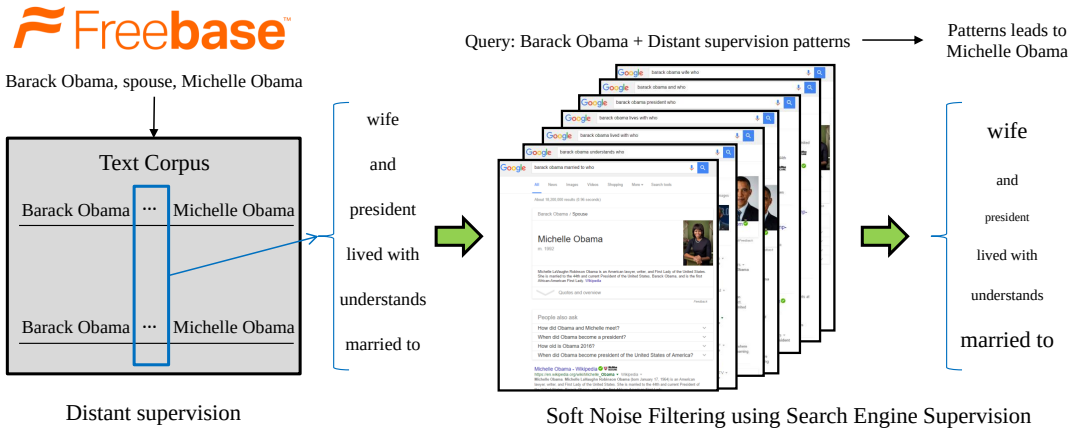

Figure 3: The distant supervision method extracts positive phrases from textual patterns between each entity pair with the relation of interest (per:spouse in this example). However, entity pair might have many different relations other than the ones we are interested in, which consist of the noise in our training data. To alleviate the problem, we examine how often we can find the second entity given the first entity and a positive phrase candidate as the query to a search engine. The noisy patterns would be weighted less in the training data because they usually cannot pass such inverse check.

classification confidence from universal schema and LSTM, we concatenate outputs from both models and remove the redundant responses. See the details of the compositional universal schema (i.e., LSTM) models in Verga et al. (2016).

Notice that we only consider the type of entities and the textual pattern between them at test time. We do not embed the entity pair because the entity pair might never be observed in the training corpus. Even if the same entity pair appears in the training data, it is hard to find provenance in the testing corpus to justify such prior knowledge.

2.2.1 Search Engine Supervision

As is the case for many knowledge base completion systems, our work relies on distant supervision to collect our training data. As we can see in the left side of Figure 3, the distant supervision training data are often noisy. In the previous year, we rely on the robustness of universal schema and the training data from RelationFactory (Roth et al., 2014), which have been cleaned based on several noise filtering methods (Roth et al., 2013). However, the problem is not completely solved because strong noise and weak signal in this task are sometimes not distinguishable, especially for a complex model like neural network.

One simple and effective way is asking human annotators to eliminate the noise or manually design noise robust features, and the effectiveness of the approach has been demonstrated in Angeli et al. (2014). However, seeking human annotations is a costly and time consuming task. This year, we seek another possibility: reduce the noises of our training data by asking search engine instead of people.

For each TAC schema, we first use few golden manual patterns indicating the relation (from RelationFactory) to select several representative entity pairs. The representative entity pairs mean that if we search the first entity + the golden manual patterns using Google, we should see the second entity very often in the top 10 retrieved web pages. For example, if we search "Barack Obama and his wife", we should get a lot of web pages containing "Michelle Obama". This implies (Barack Obama, Michelle Obama) is a good representative entity pair.

Next, we would extract many key phrase candidates from distant supervision training data of RelationFactory. For each representative entity pair and key phrase candidate, we send a query which combine the first entity and such key phrase candidate to search engine, and see how many mentions of the second entity we can find from the results. The more mentions of the second entity we found, the more likely that this key phrase candidate would imply such relation we want to extract. Finally, we emphasize the textual patterns which contain at least one verified key phrase candidate, and this can be viewed as softly removing the noises from the training data. An example could be seen in Figure 3. We test the noise removal using the TAC SF hop0 from 2012 to 2015. The F1 of Universal schema increases from 0.243 to 0.261, and the F1 of LSTM increases from 0.331 to 0.337.

2.2.2 Spanish Relation Extraction

Collecting Spanish training data is usually harder than collecting English ones, so we follow the domain transfer techniques in Verga et al. (2016). To propagate the information from English training data to Spanish, we align the rows and columns in the relation matrix of different languages as shown in Figure 2. For the rows, we perform cross-lingual entity linking on English entities and Spanish entities. For the columns, we perform word-by-word dictionary translation to enforce that the Spanish words have the same embedding as the English words with the same meaning. After the alignment, we can measure the similarity between the Spanish textual patterns and TAC schema using the compositional universal schema without preparing and cleaning the distant supervision training data for Spanish.

In an alternative viewpoint, the co-occurance of entity pairs and relations (including TAC schema and textual patterns) can be viewed as a bipartite graph. Whenever entity pair appears with a relation, there is a edge between the entity pair node and relation node. We can know a Spanish textual pattern belongs to a TAC schema because there is a path between them in this graph. During preparing for the TAC, we found that the length of path from Spanish textual patterns to TAC schema is crucial for the performance. In Verga et al. (2016), the F1 of Spanish universal schema is 0.16 for TAC SF 2012. Through decreasing the length of path, we are able to improve this F1 to 0.23 using our Spanish universal schema model.

2.2.3 Threshold Tuning

Since each system is judged by its F1 score, tuning the threshold of each relation improperly might produce a very bad performance. However, determining a good threshold is not an easy task. For some schema like gpe:residents_of_country, there are often many correct and incorrect answers in the evaluation corpus. A small fluctuation on the threshold of such type of schema might have large impact on the final performance.

Another difficulty is that we often cannot observe enough annotations in our low confidence

Table 1: TAC KBP 2016 scores for English Slot Filling (SF) and Knowledge Base (KB) construction

Run	**Method**	**Description**
SF1_ENG	SES + DS	Merging the results w/ and w/o Search Engine Supervision
SF2_ENG	SES + DS + No_Doc_Exp	As run SF1_ENG, skipping within document expansion
SF3_ENG	2015 System	The best UMass_IESL 2015 SF system
SF4_ENG	DS	Distant Supervision (i.e., w/o Search Engine Supervision)
SF5_ENG	SES	Search Engine Supervision
KB1_ENG	SES + DS	Merging the results w/ and w/o Search Engine Supervision.
KB2_ENG	DS	Distant Supervision (i.e., w/o Search Engine Supervision)
KB3_ENG	2015 System	The best UMass_IESL 2015 KB system
KB4_ENG	SES	Search Engine Supervision
KB5_ENG	KB1_ENG, ENG_Th3	As run KB1_ENG, but only tuning the thresholds by optimizing the F1 in annotated samples.

LDC max	hop0			hop1			All		
Run	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
SF1_ENG	0.2288	**0.2026**	**0.2149**	0.4444	0.0130	0.0252	0.2323	**0.1391**	**0.1740**
SF2_ENG	0.2442	0.1895	0.2134	0.1304	0.0195	0.0339	0.2342	0.1326	0.1693
SF3_ENG	**0.3023**	0.1536	0.2037	0.0811	**0.0877**	**0.0842**	0.1879	0.1315	0.1547
SF4_ENG	0.2441	0.1683	0.1992	0.5000	0.0130	0.0253	0.2488	0.1163	0.1585
SF5_ENG	0.2893	0.1503	0.1978	**0.5714**	0.0130	0.0254	**0.2954**	0.1043	0.1542
KB1_ENG	0.2709	0.1536	**0.1960**	0.0392	0.0714	0.0506	0.1278	0.1261	0.1269
KB2_ENG	0.3008	0.1209	0.1725	0.0539	0.0519	0.0529	0.1657	0.0978	0.1230
KB3_ENG	**0.3900**	0.1275	0.1921	**0.2526**	0.0779	**0.1191**	**0.3458**	0.1109	**0.1679**
KB4_ENG	0.3004	0.1193	0.1708	0.1170	0.0649	0.0835	0.2246	0.1011	0.1394
KB5_ENG	0.2110	**0.1634**	0.1842	0.0191	**0.0942**	0.0317	0.0647	**0.1402**	0.0886

outputs because TAC evaluation process only annotates the outputs from all teams and we all tend to avoid generating such low confidence responses. This means that tuning some threshold to a very low value (i.e., output nearly all candidates) might give us a better F1 score in previous years, but it would definitely give us a very bad score when every output is judged by annotators. Thus, we disallow any threshold with the value lower than 0.25 by default.

To further alleviate the problem, we use Kernel Density Estimation (KDE) to estimate the precision of our outputs under all confidences based on the annotations from TAC SF 2012 to 2015, and search the optimal thresholds according to such precision estimation and the distribution of confidences the classifiers output at TAC SF 2015. Finally, we manually increase a few thresholds which cause low precision responses at TAC KB 2015.

In the SF task, we are allowed to use different thresholds when answering hop0 and hop1 queries. Thus, when we generate the hop1 responses, we would apply higher thresholds than the ones for hop0. In the TAC, the hop1 answers would be wrong if their parent hop0 responses are wrong, so we would also apply some high thresholds on the confidence of the parent hop0 responses to reduce our false positives in our hop1 responses. All these higher thresholds are

determined using the hop1 annotations at TAC SF 2015.

3 Results and Discussion

In TAC 2016, we participate 3 language settings: English only, Spanish only, and the both languages. In each language setting, we submit 5 runs to slot filling (SF) task and 5 runs to knowledge base (KB) construction task. In each run, some combination of our modifications is turned off in order to know whether these ameliorations actually lead to a better performance when considering the provenance. The description of each run and its results are shown in Table 1 for English, Table 2 for Spanish, and Table 3 for cross-lingual setting. All scores presented in the table are micro-averages with correction for the number of entry-points (CS LDC max metric).

3.1 English Only

In Table 1, the English hop0 scores show that search engine supervision (SF5_ENG and KB4_ENG) have higher precision and similar F1 compared with the distant supervision (SF4_ENG and KB2_ENG). Furthermore, merging the results (SES+DS) of these sentence classifiers lead to a better F1 in hop0 (SF1_ENG and KB1_ENG). For the slot filling task, the within document expansion also improves the F1 overall (from 0.1693 in SF2_ENG to 0.1740 in SF1_ENG).

In the last year, predicting hop1 in SF actually brings about a worse overall F1. In this year, we tune the hop1 thresholds by optimizing the overall F1. The resulting high thresholds for hop1 cause the high precision but low recall in hop1. However, in KB task, we cannot use different thresholds for hop0 and hop1. This makes the hop1 scores in KB unstable. In some cases, whether predicting a relation instance correctly in the hop0 affects the hop1 precision dramatically. We suppose that this is why our best KB runs is actually the system from 2015.

Another interesting observation is that the evaluation process of 2015 and that of 2016 seems to score our systems very differently, especially in the KB runs, while the task uses the same set of relation schema in both years. In the TAC 2015, the overall precision of our 2015 KB system are 0.1033. However, the same system with the same parameters achieves 0.3458 for precision in TAC 2016. The difference of hop1 precision is even more dramatic (0.04 in TAC 2015 and 0.25 in TAC 2016). Being evaluated using TAC 2015 annotations, our 2016 KB system improves the hop0 F1 very significantly (from 0.19 to 0.25), but they are roughly the same in TAC 2016 evaluation.

The score differences might come from the task differences. For example, there are much more documents from discussion forum in TAC 2015. The queried relation distribution might be different at both years. The difficulty of entity linking might also changed. The difference might come from the evaluation process as well. For example, annotators use a different standard to judge the correctness of responses. It would need further investigation to identify which one is the most important factor.

3.2 Spanish Only

In Table 2, we can see that within document query expansion boosts the performance a little bit by comparing the SF1_SPA and SF2_SPA. However, adding the alias information from redirect page in Wikipedia seems to hurt the performance in Spanish by comparing the SF1_SPA and SF4_SPA. It would need further investigation to know the reason.

Table 2: TAC KBP 2016 scores for Spanish Slot Filling (SF) and Knowledge Base (KB) construction

Run	**Method**	**Description**
SF1_SPA	Default	Using redirect pages and within document query expansion
SF2_SPA	No_Doc_Exp	Using redirect pages
SF3_SPA	SPA_Th2	As run SF1, but allowing lower thresholds
SF4_SPA	No_Redirect	Using within document query expansion
SF5_SPA	SPA_Th2 + No_Doc_Exp	As run SF2, but allowing lower thresholds
KB1_SPA	Emb_Link + SPA_Th2	Using embedding linker and allowing lower thresholds
KB2_SPA	Emb_Link	Using embedding linker
KB3_SPA	Alias_Link	Using alias linker
KB4_SPA	KB1, No_Inv_Check	As run KB1, but turning off inverse check
KB5_SPA	Alias_Link + SPA_Th2	Using alias linker and allowing lower thresholds

LDC max	hop0			hop1			All		
Run	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
SF1_SPA	0.1327	0.2691	0.1777	0.0000	0.0000	0.0000	0.1327	0.1787	0.1523
SF2_SPA	0.1323	0.2369	0.1698	0.0000	0.0000	0.0000	0.1323	0.1573	0.1437
SF3_SPA	0.0419	**0.3333**	0.0745	0.0000	0.0000	0.0000	0.0419	**0.2213**	0.0705
SF4_SPA	**0.1387**	0.2851	**0.1866**	0.0000	0.0000	0.0000	**0.1387**	0.1893	**0.1601**
SF5_SPA	0.0427	0.3213	0.0754	0.0000	0.0000	0.0000	0.0427	0.2133	0.0712
KB1_SPA	0.2698	0.0683	0.1090	0.0000	0.0000	0.0000	0.1288	0.0453	0.0671
KB2_SPA	0.2743	**0.1245**	**0.1713**	0.0409	**0.0556**	0.0471	0.1338	**0.1013**	0.1153
KB3_SPA	0.2804	0.1205	0.1685	**0.0600**	0.0476	**0.0531**	0.1739	0.0960	**0.1237**
KB4_SPA	0.2742	0.0683	0.1093	0.0000	0.0000	0.0000	0.1298	0.0453	0.0672
KB5_SPA	**0.2963**	0.0643	0.1056	0.0000	0.0000	0.0000	**0.2192**	0.0427	0.0714

It is difficult to tune the thresholds for Spanish sentence classifiers because we don't have annotations for the inverse relations and hop1 responses in the TAC 2012 and Spanish pilot runs for 2016. Thus, we prepare 2 different thresholds and skip all hop1 queries in the SF task. During threshold tuning, we found that lower the optimal thresholds for TAC 2012 could give us higher recall with similar precision in the pilot run this year. Thus, we try two ways to achieve the goal. The first way is to disallow any threshold to go below 0.25, and multiply all the resulting thresholds by 0.7. The second way is to disallow any threshold to be lower than 0.1.

From the results, we see that the runs using the first threshold tuning way (SF1_SPA, SF2_SPA, SF4_SPA, KB2_SPA, KB3_SPA) perform much better than the ones using the second way (SF3_SPA, SF5_SPA, KB1_SPA, KB4_SPA, KB5_SPA). This indicates that allowing really low thresholds might be dangerous, even though the it could give you a better F1 in the partially observable annotations at previous years. The classifier thresholds used for our KB and SF are roughly the same. However, our KB gets high precision but low recall results, while our SF generates high recall but low precision responses. It might suggest that our Spanish query expansion generates too many noisy aliases or the Spanish entity linking over-split the clusters.

Table 3: TAC KBP 2016 scores for English and Spanish Slot Filling (SF) and Knowledge Base (KB) construction. To simplify the descriptions, we use the same abbreviations in the method column of Table 1 and Table 2. The parenthesis indicates the setting is applied to which language.

Run	**Method**	**Description**
SF1_XLING	Eng1 + Spa1	Doc_Exp + Redirect
SF2_XLING	Eng2 + Spa2	Redirect
SF3_XLING	Eng1 + Spa3	Doc_Exp + Redirect + SPA_Th2 (SPA)
SF4_XLING	Eng2 + Spa1	Doc_Exp (SPA) + Redirect
SF5_XLING	Eng4 + Spa2	Doc_Exp (ENG) + Redirect + No_SES (ENG)
KB1_XLING	Eng1 + Spa1	SPA_Th2 (SPA)
KB2_XLING	Eng4 + Spa2	No_DS (ENG)
KB3_XLING	Eng1 (ENG_Th2) + Spa2	ENG_Th2 (ENG)
KB4_XLING	Only USchema	No_LSTM (ENG) + No_LSTM (SPA)
KB5_XLING	Only LSTM	No_USchema (ENG) + No_USchema (SPA)

LDC max	hop0			hop1			All		
Run	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
SF1_XLING	0.1801	0.1572	0.1678	**0.2857**	0.0066	0.0128	0.1815	0.1067	**0.1343**
SF2_XLING	**0.1921**	0.1439	**0.1645**	0.1304	**0.0098**	**0.0183**	**0.1891**	0.0990	0.1299
SF3_XLING	0.0814	**0.1803**	0.1122	0.2000	0.0066	0.0127	0.0823	**0.1220**	0.0983
SF4_XLING	0.1775	0.1514	0.1634	0.1333	**0.0098**	**0.0183**	0.1757	0.1039	0.1306
SF5_XLING	0.1869	0.1340	0.1561	**0.2857**	0.0066	0.0128	0.1884	0.0913	0.1230
KB1_XLING	0.2854	0.0951	0.1427	0.0384	0.0377	0.0380	0.1377	0.0759	0.0978
KB2_XLING	0.2772	0.0844	0.1294	0.0716	0.0410	0.0521	0.1771	0.0698	0.1002
KB3_XLING	0.2537	**0.1141**	**0.1574**	0.0591	**0.0557**	**0.0574**	0.1537	**0.0946**	**0.1171**
KB4_XLING	**0.4588**	0.0323	0.0603	**0.1667**	0.0066	0.0126	**0.3945**	0.0236	0.0446
KB5_XLING	0.3406	0.0778	0.1266	0.1091	0.0295	0.0465	0.2540	0.0616	0.0991

3.3 English and Spanish

In this year, we only process English and Spanish corpus and queries, but correct Chinese responses from other teams are also considered while computing the recall. This explains the generally low recall in Table 3. In most of the submissions, we just perform cross-lingual entity linking or query expansion in a English run and a Spanish run, and the method column in Table 3 indicates which runs we choose. The KB4_XLING and KB5_XLING are exceptions because we want to compare the performances of only using universal schema and only using LSTM, while all other runs are the combination of these two sentence classifiers. By comparing these runs, we can know that LSTM (KB5_XLING) has much higher recall than universal schema (KB4_XLING), and combine these two plus the search engine supervision can further improve the performance (e.g., KB3_XLING).

In the English only and cross-lingual submissions, we test 3 different threshold tuning methods: optimizing the F1 in annotated samples (i.e., traditional method), estimating precision in each threshold using KDE, and manual fine tuning to increase the precision at the end. In the final English KB runs (i.e., KB5_ENG), we only optimize the F1 in annotated sam-

ples. It leads to worse scores compared with KB1_ENG. Nevertheless, turning off final manual fine tuning in KB3_XLING does not result in significant drop in the precision. This might imply that the KDE precision estimation is helpful for threshold tuning process but the manual fine tuning at the end is unnecessary.

4 Conclusion

Motivated by the scarcity of language resources, we proposed several ways to hurdle the challenges, including transfer learning and automatically collecting more online resources to remove noises from distant supervision training data. The scores of various runs verify that the most of our improvements have positive effects on our relation extraction system.

The evaluation scores in TAC 2016 also bring out some interesting questions. For instance, TAC 2016 scores our 2015 KB systems much higher than TAC 2015 did. In addition, a few resources such as Wikipedia redirect pages do not always lead to a better performance, and some resources such as the ones collected from the search engine improve the system performance but not very much. This motivates us to deepen the understanding of universal schema and our relation extraction system in the future.

References

Adel, H., Roth, B., and Schütze, H. (2016). Comparing convolutional neural networks to traditional models for slot filling. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Angeli, G., Gupta, S., Premkumar, M. J., Manning, C. D., Ré, C., Tibshirani, J., Wu, J. Y., Wu, S., and Zhang, C. (2014). Stanfords distantly supervised slot filling systems for KBP 2014. In *Text Analysis Conference (TAC-KBP)*.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD international conference on Management of data*.

Francis-Landau, M., Durrett, G., and Klein, D. (2016). Capturing semantic similarity for entity linking with convolutional neural networks. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Huang, H., Heck, L., and Ji, H. (2015). Leveraging deep neural networks and knowledge graphs for entity disambiguation. *arXiv preprint arXiv:1504.07678*.

McCallum, A., Schultz, K., and Singh, S. (2009). FACTORIE: Probabilistic programming via imperatively defined factor graphs. In *Neural Information Processing Systems (NIPS)*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*.

Nguyen, T. H., Fauceglia, N., Muro, M. R., Hassanzadeh, O., Gliozzo, A. M., and Sadoghi, M. (2016). Joint learning of local and global features for entity linking via neural networks. In *Proceedings of COLING*.

Passos, A., Kumar, V., and McCallum, A. (2014). Lexicon infused phrase embeddings for named entity resolution. In *Conference on Natural Language Learning (CoNLL)*.

Riedel, S., Yao, L., McCallum, A., and Marlin, B. M. (2013). Relation extraction with matrix factorization and universal schemas.

In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Roth, B., Barth, T., Chrupala, G., Gropp, M., Klakow, D., Wintner, S., Goldwater, S., and Rielzler, S. (2014). RelationFactory: A fast, modular and effective system for knowledge base population. In *European Chapter of the Association for Computational Linguistics (EACL)*.

Roth, B., Barth, T., Wiegand, M., and Klakow, D. (2013). A survey of noise reduction methods for distant supervision. In *Proceedings of the 2013 workshop on Automated knowledge base construction*.

Roth, B., Monath, N., Belanger, D., Strubell, E., Verga, P., and McCallum, A. (2015). Building knowledge bases with universal schema: Cold start and slot-filling approaches. In *Text Analysis Conference (TAC-KBP)*.

Sun, Y., Lin, L., Tang, D., Yang, N., Ji, Z., and Wang, X. (2015). Modeling mention, context and entity with neural networks for entity disambiguation. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Verga, P., Belanger, D., Strubell, E., Roth, B., and McCallum, A. (2016). Multilingual relation extraction using compositional universal schema. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Yamada, I., Shindo, H., Takeda, H., and Takefuji, Y. (2016). Joint learning of the embedding of words and entities for named entity disambiguation. In *Conference on Computational Natural Language Learning (CoNLL)*.