Building Knowledge Bases with Universal Schema: Cold Start and Slot-Filling Approaches

Benjamin Roth **Nicholas Monath** **David Belanger**
Emma Strubell **Patrick Verga** **Andrew McCallum**
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA, 01003, USA
beroth@cs.umass.edu

Abstract

We compare the performance of two different relation prediction architectures based on the same relation predictors. The knowledge base construction architecture builds a complete knowledge base for the entire corpus, and commits to entity linking and clustering decisions ahead of time. The query-driven slot filling architecture can make entity expansion and retrieval decisions on the fly, and has the flexibility to trade precision for recall. We use a wide range of established and novel techniques for our relation extraction components. They include distant supervision-based classifiers (SVM and convolutional neural nets), rule-based extractors, and semi-supervised matrix embedding methods taking into account all co-occurrences of surface patterns and entities in the corpus (universal schema).

1 Overview

UMass IESL participated in both Cold Start tasks: KB construction and Slot Filling. While the relation prediction relies on the same models for both tasks, we have developed different, task-dependent system architectures for each setting. The KB construction task requires a complete KB to be built ahead of time. This includes clustering the entire set of entity mentions into disambiguated KB entities, and connecting the entities by predicted relations. For the Slot-Filling (SF) Cold Start setting, the knowledge base has to be constructed only partially at query time, starting from the specified query entities. The SF setting is less rigid than the KB setting. Since the entity mentions are not pre-clustered, entity expansion techniques are query centered and leave more room for controlling precision and recall. Since it has been shown that current Slot-Filling systems mainly suffer from low recall, this may be a desirable property. On the other hand, having a complete, query-independent knowledge base (as in the KB construction setting) may open new avenues for joint reasoning, knowledge discovery and filtering.

Which of the settings is more appropriate in a real-world scenario will depend on the particular circumstances. It is, however, interesting to understand what the exact tradeoffs are between the two settings. Having access to two different high-level architectures that use the same relation predictors, the UMass IESL runs offer a comparison of the impact of both settings on the final results.

2 Relation Classification Techniques

All our runs, both in the cold start construction setting as well as in the slot-filling setting, are based on the following relation predictors:

- **Universal Schema** (Riedel et al., 2013): We collected the surface patterns between all entity pairs in the TAC2014 corpus, and collected distant supervision signals for a subset of these entity pairs from Freebase (Bollacker et al., 2008). (The Freebase relations were manually mapped to TAC relations.) We represented this data as a matrix with the entity pairs as the rows, and with patterns and relations as columns. We factorized this matrix and embedded entity pairs, relations and patterns in a 100-dimensional vector space. We then construct a list of text patterns that are predictive of TAC relations with high confidence by finding patterns whose vector embedding has high cosine similarity with the embedding of a TAC relation. We then tuned confidence thresholds using previous TAC data. At test time, facts were predicted wherever one of our high-confidence patterns occured.

- **Convolutional Neural Networks (CNNs):** We collected distant supervision data from the TAC 2014 corpus and Freebase and trained a convolutional neural network on the sentences (where the entities were wildcarded by special tokens *ARG1* and *ARG2*). Convolutional neural networks provide a simple, intuitive method for producing deep representations of text (Collobert et al., 2011; Kim, 2014; Kalchbrenner et al., 2014). The first layer is a lookup table of word embeddings, and then convolutions are applied across the time axis. These act as 'soft' n-grams. To obtain a single sentence-level representation, information is pooled across the time axis. CNNs have been applied to relation extraction tasks recently (Zeng et al., 2014, 2015). See Zhang and Wallace (2015) for practical recommendations for applying CNNs to text.

- **SVMs** (Roth et al., 2013): On the same data we trained a set of binary SVMs, one for each relation.

- **Manual rules:** We used the set of prespecified rules from the RelationFactory (Roth et al., 2014a) relation extraction system.

3 Cold Start KB Construction

For cold start construction, the entity mentions need to be clustered, i.e. each mention needs to be assigned to an entity in the KB. In exploratory experiments on the TAC 2014 data we compared several clustering and linking techniques such as perfect string match (baseline), similarity based on contexts (using IR metrics) and based on word-embeddings (Mikolov et al., 2013a,b). We found that an approach based on context for linking, and based on surface forms for unlinkable mention clustering is robust and works well compared to the other explored options, and we use this approach described below for our submitted runs.

The entity linking algorithm first performs within-document corefence and selects a canonical mention for each resulting within-document entity. This canonical mention is

	submission			not predicting 2-hop queries		
Run	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
SF1	**0.2392**	0.1463	**0.1816**	0.3252	0.1239	0.1795
SF2	0.0966	0.1655	0.1220	0.2137	**0.1368**	*0.1668*
SF3	0.2136	0.1521	0.1777	0.2996	0.1301	***0.1814***
SF4	0.2285	0.1188	0.1564	**0.3583**	0.1038	*0.1609*
SF5	0.2106	0.1344	0.1641	0.2981	0.1136	*0.1645*
KB1	0.1111	0.1526	0.1286	0.2557	0.1208	*0.1641*
KB2	0.1069	**0.1737**	0.1324	0.1946	0.1332	*0.1582*
KB3	0.1019	0.1222	0.1111	0.213	0.0950	*0.1314*
KB4	0.1305	0.1304	0.1304	0.2357	0.1043	*0.1446*

Table 1: Scores for hop1 and hop2 combined. Scores are micro-averages with correction for the number of entry-points (CS LDC max metric). *Left:* Scores of runs as submitted. *Right:* Scores of runs with all hop2 predictions removed (but scored as before). Increased F1 by not predicting hop2 relations is marked in italic.

Run	**Prec hop1, hop2**	**Rec hop1, hop2**	**F1 hop1, hop2**
SF1	0.3252, **0.0952**	0.1893, 0.0637	**0.2393**, **0.0764**
SF2	0.2137, 0.0269	**0.2090**, 0.0833	0.2113, 0.0407
SF3	0.2996, 0.0796	0.1987, 0.0640	0.2390, 0.0710
SF4	**0.3583**, 0.0635	0.1585, 0.0425	0.2198, 0.0509
SF5	0.2981, 0.0793	0.1735, 0.0592	0.2193, 0.0678
KB1	0.2557, 0.0347	0.1845, 0.0910	0.2144, 0.0502
KB2	0.1946, 0.0433	0.2035, **0.1176**	0.1989, 0.0633
KB3	0.2130, 0.0363	0.1451, 0.0789	0.1726, 0.0497
KB4	0.2357, 0.0471	0.1593, 0.0759	0.1901, 0.0582

Table 2: Scores broken down into hop1 and hop2. Scores are micro-averages with correction for the number of entry-points (CS LDC max metric).

used for linking (or NIL clustering) all other mentions of its cluster. Using Wikipedia articles, anchor text, and Freebase, we link each mention to a Freebase entity whenever possible. The unlinkable mentions are clustered using their surface forms and entity types assigned by the tagger.

For every canonical entity mention, a list of Wikipedia articles reachable via link anchor text is retrieved. Freebase is used to check that the Wikipedia articles are for entities of the respective type. Of those articles, the one with the highest cosine similarity to the context around the entity mention is selected (using a sparse bag-of-words representation), and the entity mention is linked to the article whenever the similarity exceeds a threshold. Otherwise a new entity is created, represented by the surface string and type of the entity mention.

We submitted four KB Construction runs:

- **KB1 (SVM+USchema):** Run based on SVM predictors and surface patterns obtained from matrix factorization on the

TAC 2014 corpus (universal schema).

- **KB2 (all modules):** SVM predictors, universal schema patterns, convolutional neural networks, and manual rules.
- **KB3 (KB1+inverse check):** As run KB1, but enforcing type and number constraints on inverse relations.
- **KB4 (KB2+inverse check):** As run KB2, but enforcing type and number constraints on inverse relations.

4 Cold Start Slot Filling

The Slot-Filling Cold Start system is less rigid in assigning the query to textual mentions. Since the query and its type is presented, the system can assume it is an entity, and does not need to rely on a named entity tagger for detecting its mentions and type. In the Slot-Filling setting, we rely on RelationFactory (Roth et al., 2014a) for retrieving the entities and relation contexts. For each query entity, two lists of aliases bases on anchor text statistics are created: A large list, using *frequent* co-occurring aliases, and a shorter sub-list of those only containing *highly correlated* aliases. The shorter list is used to retrieve relevant documents for the query, while the larger list is used to match query mentions in those documents. Note that this mechanism allows for controlling both precision and recall effects.

The short characterization of our Slot-Filling runs is as follows:

- **SF1 (RF+SVM+USchema):** This is based on the standard RelationFactory modules (SVM, manual rules, alternate names) and surface patterns, obtained from matrix factorization on the TAC 2014 corpus.
- **SF2 (high recall):** As in run SF1, additionally with a predictor based on convolutional neural networks (CNN). To further increase recall, entity types are ignored, i.e. any tagged entity is a potential argument for all slots.
- **SF3 (RF+CNN+USchema):** As in run SF2, but the SVM replaced by the CNN module.
- **SF4 (2014 system):** This is the UMass IESL 2014 system (Roth et al., 2014b) with two minor changes: we use a new NE tagger (based on Factorie) and we added the inverse cold start relations to the SVM predictor.
- **SF5 (KB equiv):** This is the same as SF1, but without the alternate names module and the manual rules module. With respect to the relation prediction components, this run is equivalent to run KB1 of the KB construction task. (However, the retrieval pipeline and entity matching are very different between the systems for the two tasks).

Since it is more difficult to obtain high recall than high precision, relation extraction systems typically have a bias towards precision. Therefore, it seems that increasing recall would profit the overall F1 score most. We ran several experiments to further increase the recall of our system. Specifically we tried to find more entity candidates for the slot filler argument positions by using the following entity mention detection mechanisms instead of standard NE-tagging:

- Ignoring the type information of the NE-tags.
- Noun chunking.

- POS-tagging and keeping noun sequences only.

Interestingly, all these methods could only slightly increase recall; at the same time they had a strong negative impact on precision, and therefore hurt the overall performance.

5 Discussion

Looking at the *submission* scores in Table 1, it can be observed that for most runs precision is higher than recall. This is in line with most (but not all) submissions by other teams. It seems, therefore, that the biggest potential for improvement would lie in increasing system recall.

Our run *SF2* however shows that this can be a tricky endeavor, since the cold start setting is extremely sensitive to drops in precision for hop2 queries (see Table 2). The reason for this is that for longer chains of relations, prediction accuracy (making independent predictions for all relations) decreases quadratically in the number of hops. By returning as many responses as possible in run *SF2*, we traded a gain in recall for a loss in precision. While for the hop1 queries, this decreased the overall score by about 12% relative F1 (from 0.2393 to 0.2113), the performance loss for hop2 queries was about 47% relative F1 (from 0.0764 to 0.0407).

This lead us to an experiment summarized in the right part of Table 2. We investigated whether, due to the quadratic drop of precision for the second hop, it is beneficial to predict hop2 relations at all. Astonishingly enough, *not* predicting hop2 relations increased the overall F1 in 8 out of 9 cases.

This surprising outcome points to the specific challenge of jointly predicting chains of relations. Our submitted system used independent relation predictors and did not directly address this issue. In light of the above analysis, it would be interesting to combine the universal schema cold start system with a reasoning component as proposed in Neelakantan et al. (2015).

Comparing the slot-filling and KB runs that use the same prediction modules, run *SF5* and run *KB1*, it is interesting to see that the slot-filling run has higher precision (and slightly lower recall) than the KB run. This is surprising since the slot-filling run employs a dedicated query-expansion mechanism to increase recall, while the KB system links detected entity mentions to Wikipedia articles ahead of time, irrespective of the current query.

We hypothesize that in the KB setting, linking entity mentions to Wikipedia (by using anchor text and document context) may yield lower precision when mentions of infrequent entities (not in Wikipedia) are wrongly linked to a more frequent Wikipedia entity. Moreover, unlinkable named entities are clustered only based on their surface form and type. This may hurt precision when different entities with the same surface form are linked together. In the SF setting however, restricting the query to match only entities in the retrieved relevant documents seems to have a greatly positive impact on precision.

6 Conclusion

We haven given an overview of the UMass IESL approaches to full vs. query-based knowledge base construction, and we demonstrated the combination of a range of relation prediction components. While most of the prediction components are based on distant supervision, universal schema additionally allows for making use of more corpus-level co-occurrences (not just of entities occurring in a training KB).

We highlighted some observations regarding specific challenges in the cold start setting for predicting relational chains with more than one hop, and we highlighted that these queries are particularly sensitive to the precision of the relational predictors. Our findings motivate more research into reasoning or modeling joint prediction for chains of relations.

References

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *EMNLP*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Neelakantan, A., Roth, B., and McCallum, A. (2015). Compositional vector space models for knowledge base completion. In *NAACL '15*.

Riedel, S., Yao, L., McCallum, A., and Marlin, B. M. (2013). Relation extraction with matrix factorization and universal schemas. In *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '13)*.

Roth, B., Barth, T., Chrupała, G., Gropp, M., and Klakow, D. (2014a). Relationfactory: A fast, modular and effective system for knowledge base population. *EACL 2014*, page 89.

Roth, B., Barth, T., Wiegand, M., Singh, M., and Klakow, D. (2013). Effective slot filling based on shallow distant supervision methods. In *Text Analysis Conference (TAC 2013)*.

Roth, B., Strubell, E., Sullivan, J., Vikraman, L., Silverstein, K., and McCallum, A. (2014b). Universal Schema for Slot-Filling, Cold-Start KBP and Event Argument Extraction: UMassIESL at TAC KBP 2014. In *Text Analysis Conference (Knowledge Base Population Track) '14 Workshop (TAC KBP)*, Gaithersburg, Maryland, USA.

Zeng, D., Liu, K., Chen, Y., and Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. *EMNLP*.

Zeng, D., Liu, K., Lai, S., Zhou, G., and Zhao, J. (2014). Relation classification via convolutional deep neural network. In *Proceedings of COLING*, pages 2335–2344.

Zhang, Y. and Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.