Multilingual Relation Extraction using Compositional Universal Schema

Anonymous NAACL submission

Abstract

When building a knowledge base (KB) of entities and relations from multiple structured KBs and text, *universal schema* represents the union of all input schema, by jointly embedding all relation types from input KBs as well as textual patterns expressing relations. In previous work, textual patterns are parametrized as a single embedding, preventing generalization to unseen textual patterns. In this paper we employ an LSTM to compositionally capture the semantics of relational text. We dramatically demonstrate the flexibility of our approach by evaluating in a multilingual setting, in which the English training data entities overlap with the seed KB, but the Spanish text does not. Additional improvements are obtained by tying word embeddings across languages. In extensive experiments on the English and Spanish TAC KBP benchmark, our techniques provide substantial accuracy improvements. Furthermore we find that training with the additional non-overlapping Spanish also improves English relation extraction accuracy. Our approach is thus suited to broad-coverage automated knowledge base construction in low-resource languages and domains.

1 Introduction

The goal of automatic knowledge base construction (AKBC) is building a structured knowledge base (KB) of facts using a noisy corpus of raw text evidence, and perhaps an initial seed KB to be augmented (**???**). AKBC supports downstream reasoning at a high level about extracted entities and their relations, and thus has broad-reaching applications to a variety of domains.

One challenge in AKBC is aligning knowledge from a structured KB with a text corpus in order to perform supervised learning through *distant supervision*. *Universal schema* (**?**) along with its extensions (**?????**) avoids this issue of alignment by jointly embedding KB relations, entities and text patterns. This allows information to propagate between KB annotation and corresponding textual evidence without explicit sentence-relation alignment.

Previous approaches to universal schema express each text relation as a distinct item to be embedded. This harms its ability to generalize, making it impossible to process inputs not precisely seen at training time. However, for large-scale applications we are interested in generalizing to new text patterns, new entities, and even new domains. We focus on the extreme example of domain adaptation to a completely new language, which may have limited resources or labeled data such as treebanks, and only rarely a KB with adequate coverage.

This paper leverages universal schema to train a deep sentence encoder that captures the compositional semantics of textual relations, allowing for prediction on inputs never seen before. We dramatically demonstrate the generality of our method by evaluating in a multilingual transfer learning setting, extracting relations from a corpus in a new language with no coverage in an existing KB, requiring only that the entities in the text corpora for two languages overlap, as depicted in Figure 1.

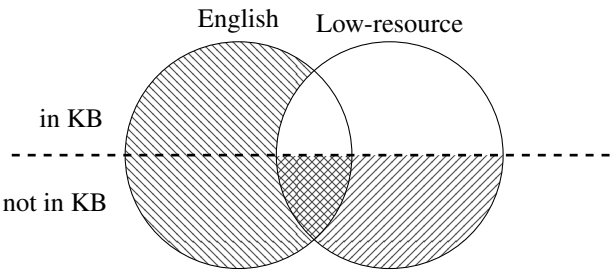
We further improve our models by tying a small set of word embeddings across languages using only simple knowledge about word-level translations, learning to embed semantically similar textual patterns from different languages into the same latent space.

In extensive experiments on the TAC Knowledge Base Population (KBP) slot-filling benchmark we perform relation extraction in Spanish with no labeled data or direct Spanish-KB overlap, demonstrating that our approach is well-suited for broad-coverage AKBC in low-resources languages and domains. Interestingly, we also find that joint training with Spanish improves English accuracy.

2 Background

AKBC extracts unary attributes of the form (*subject*, *attribute*), typed binary relations of the form (*subject*, *rela-*

Figure 1: Splitting the entities in a multilingual AKBC training set into parts. We only require that entities in the two corpora overlap. Remarkably, we can train a model for the low-resource language even if entities in the low-resource language do not occur in the KB.



tion, *object*), or higher-order relations. We refer to subjects and objects as *entities*. This work focuses solely on extracting binary relations, though many of our techniques generalize naturally to unary prediction. Generally, for example in Freebase (**?**), higher-order relations are expressed in terms of collections of binary relations.

We now describe prior work on approaches to AKBC. They all aim to predict (*s, r, o*) triples, but differ in terms of: (1) input data leveraged, (2) types of annotation required, (3) definition of relation label schema, and (4) whether they are capable of predicting relations for entities unseen in the training data. Note that all of these methods require pre-processing to detect entities, which may result in additional KB construction errors.

2.1 Relation Extraction as Link Prediction

A knowledge base is naturally described as a graph, in which entities are nodes and relations are labeled edges (**??**). In the case of *knowledge graph completion*, the task is akin to link prediction, assuming an initial set of (*s, r, o*) triples. See **?** for a review. No accompanying text data is necessary, since links can be predicted using properties of the graph, such as transitivity. In order to generalize well, prediction is often posed as low-rank matrix or tensor factorization. A variety of model variants have been suggested, where the probability of a given edge existing depends on a multi-linear form (**??????**), or non-linear interactions between s, r, and o (**?**).

2.2 Relation Extraction as Sentence Classification

Here, the training data consist of (1) a text corpus, and (2) a KB of seed facts with provenance, ie. supporting evidence, in the corpus. Given individual an individual sentence, and pre-specified entities, a classifier predicts whether the sentence expresses a relation from a target schema. To train such a classifier, KB facts need to be aligned with supporting evidence in the text, but this is often challenging. For example, not all sentences containing Barack and Michelle Obama state that they are married. A variety of one-shot and iterative methods have addressed the alignment problem (**????????**). An additional degree of freedom in these approaches is whether they classify individual sentences or predicting at the corpus level by aggregating information from all sentences containing a given pair of entities before prediction. The former approach is often preferrable in practice, due to the simplicity of independently classifying individual sentences and the ease of associating each prediction with a provenance.

2.3 Open-Domain Relation Extraction

In the previous two approaches, prediction is carried out with respect to a fixed schema R of possible relations r. This may overlook salient relations that are expressed in the text but do not occur in the schema. In response, *open-domain* information extraction (OpenIE) lets the text speak for itself: R contains all possible patterns of text occuring between entities s and o (**???**). These are obtained by filtering and normalizing the raw text. The approach offers impressive coverage, avoids issues of distant supervision, and provides a useful exploratory tool. On the other hand, OpenIE predictions are difficult to use in downstream tasks that expect information from a fixed schema.

Table 1 provides examples of OpenIE patterns. The examples in row two and three illustrate relational contexts for which similarity is diffult to be captured by an OpenIE approach because of their syntactically complex constructions. This motivates the technique in Section 4, which uses a deep architecture applied to the raw tokens, instead of rigid rules for normalizing text to obtain patterns.

2.4 Universal Schema

When applying Universal Schema (**?**) (USchema) to relation extraction, we combine the OpenIE and link-prediction perspectives. By jointly modeling both OpenIE patterns and the elements of a target schema, the method captures broader relational structure than multi-class classification approaches that just model the target schema. Furthermore, the method avoids the distant supervision alignment difficulties of Section 2.2.

? augment a knowledge graph from a seed KB with additional edges corresponding to OpenIE patterns observed in the corpus. Even if the user does not seek to predict these new edges, a joint model over all edges may be able to exploit regularities of the OpenIE edges to improve modeling of the labels from the target schema.

Table 1: Examples of sentences expressing relations. Context tokens (italicized) consist of the text occurring between entities (bold) in a sentence. OpenIE patterns are obtained by normalizing the context tokens using hand-coded rules. The top example expresses the per:siblings relation and the bottom two examples both express the per:cities_of_residence relation.

Sentence (context tokens italicized)	OpenIE pattern
Khan *'s younger sister,* **Annapurna Devi**, who later married Shankar, developed into an equally accomplished master of the surbahar, but custom prevented her from performing in public.	*arg1*'s * sister *arg2*
A professor emeritus at Yale, **Mandelbrot** *was born in Poland but as a child moved with his family to* **Paris** where he was educated.	*arg1* * moved with * family to *arg2*
Kissel *was born in Provo, Utah, but her family also lived in* **Reno**.	*arg1* * lived in *arg2*

The data still consist of (s, r, o) triples, which can be predicted using link-prediction techniques such as low-rank factorization. **?** explore a variety of approximations to the 3-mode (s, r, o) tensor. One such probabilistic model is:

$$\mathbb{P}\left((s, r, o)\right) = \sigma\left(u_{s,o}^{\top} v_r\right), \tag{1}$$

where $\sigma()$ is a sigmoid function, $u_{s,o}$ is an embedding of the entity pair (s, o), and v_r is an embedding of the relation r, which may be an OpenIE pattern or a relation from the target schema. All of the exposition and results in this paper use this factorization, though many of the modeling techniques we present later could be applied easily to the other factorizations described in **?**. Note that learning unique embeddings for each OpenIE relations does not guarantee that similar patterns, such as the final two in Table 1, will be embedded similarly.

As with most of the techniques in Section 2.1, the data only consist of positive examples of edges. The absence of an annotated edge does not imply that the edge is false. In fact, we seek to predict some of these missing edges as true. **?** employ the Bayesian Personalized Ranking (BPR) approach of **?**, which does not explicitly model unobserved edges as negative, but instead seeks to rank the probability of observed triples above unobserved triples.

Recently, **?** extended USchema to not learn individual pattern embeddings v_r, but instead to embed text patterns using a deep architecture applied to word tokens. This shares statistical strength between OpenIE patterns with similar words. We leverage this approach in Section 4. Additional work has modeled the regularities of multi-hop paths through knowledge graph augmented with text patterns (**????**).

3 Training a Sentence Classifier without Alignment

Similar to many link prediction approaches, (**?**) perform transductive learning, where a model is learned jointly over train and test data. Predictions are made by using the model to identify edges that were unobserved in the test data but likely to be true. The approach is vulnerable to the *cold start* problem in collaborative filtering (**?**): it is unclear how to form predictions for unseen entity pairs, without re-factorizing the entire matrix or applying heuristics.

In response, this paper re-purposes USchema as a means to train a sentence-level relation classifier, like those in Section 2.2, which allows us to avoid errors from aligning distant supervision to the corpus. It provides improved accuracy, is more deployable for real-world applications, and provide opportunities in Section 5 to improve multilingual AKBC.

We produce predictions using a very simple approach: (1) scan the corpus and extract a large quantity of triplets (s, r_{text}, o), where r_{text} is an OpenIE pattern. For each triplet, if the similarity between the embedding of r_{text} and the embedding of a target relation r_{schema} is above some threshold, we predict the triplet $(s, r_{\text{schema}}, o)$, and its provenance is the input sentence containing (s, r_{text}, o). We refer to this technique as *pattern scoring*. In our experiments, we use the cosine distance between the vectors. In Section A.2, we discuss details for how to make this distance well-defined.

4 Predictions for Unseen Text Patterns

The pattern scoring approach is subject to an additional cold start problem: input data may contain patterns unseen in training. This section describes a method for using USchema to train a relation classifier that can take arbitrary context tokens (Section 2.3) as input.

Fortunately, the cold start problem for context tokens is more benign than that of entities since we can exploit statistical regularities of text: similar sequences of context tokens should be embedded similarly. Therefore, following **?**, we embed raw context tokens compositionally using a deep architecture. Unlike **?**, this requires no manual rules to map text to OpenIE patterns and can embed any possible input string. The modified USchema likelihood is:

$$\mathbb{P}\left((s, r, o)\right) = \sigma\left(u_{s,o}^{\top} \text{Encoder}(r)\right). \tag{2}$$

Here, if r is raw text, then $\text{Encoder}(r)$ is parametrized by a deep architecture. If r is from the target schema, $\text{Encoder}(r)$ is a produced by a lookup table (as in traditional USchema). Though such an encoder increases the computational cost of test-time prediction over straightforward pattern matching, evaluating a deep architecture can be done in large batches in parallel on a GPU.

Both convolutional networks (CNNs) and recurrent networks (RNNs) are reasonable encoder architectures, and we consider both in our experiments. CNNs have been useful in a variety of NLP applications (**???**). Unlike **?**, we also consider RNNs, specifically Long-Short Term Memory Networks (LSTMs) (**?**). LSTMs have proven successful in a variety of tasks requiring encoding sentences as vectors (**??**). In our experiments, LSTMs outperform CNNs.

There are two key differences between our sentence encoder and that of **?**. First, we use the encoder at test time, since we process the context tokens for held-out data. On the other hand, **?** adopt the transductive approach where the encoder is only used to help train better representations for the relations in the target schema; it is ignored when forming predictions. Second, we apply the encoder to the raw text between entities, while **?** first perform syntactic dependency parsing on the data and then apply an encoder to the path between the two entities in the parse tree. We avoid parsing, since we seek to perform multilingual AKBC, and many languages lack linguistic resources such as treebanks. Even parsing non-newswire English text, such as tweets, is extremely challenging.

Prior work has applied deep learning to small-scale relation extraction problem, where functional relationships are detected between common nouns. **?** apply an LSTM to a parse path, while **?** use a CNN on the raw text, with a special temporal pooling operation to separately embed the text around each entity.

4.1 Modeling Frequent Text Patterns

Despite the coverage advantages of using a deep sentence encoder, separately embedding each OpenIE pattern, as in **?**, has key advantages. In practice, we have found that many high-precision patterns occur quite frequently. For these, there is sufficient data to model them with independent embeddings per pattern, which imposes minimal restrictions on the relationship between embeddings. On the other hand, Some discriminative phrases are idiomatic, i.e.. their meaning is not constructed compositionally from their constituents. For these, the inductive bias of a sentence encoder is inappropriate.

Therefore, using pattern embeddings and deep token-based encoders have very different strengths and weaknesses. One values specificity, and models the head of the text distribution well, while the other has high coverage and captures the tail. In our experiments, we demonstrate that the lookup table approach outperforms using an encoder, but that an ensemble of both models performs substantially better than either in isolation.

5 Multilingual Relation Extraction with Zero Annotation

The models described in previous two sections provide broad-coverage relation extraction that can generalize to all possible input entities and text patterns, while avoiding error-prone alignment of distant supervision to a corpus. Next, we describe techniques for an even more challenging generalization task: relation classification for input sentences in completely different languages.

Training a sentence-level relation classifier, either using the alignment-based techniques of Section 2.2, or the alignment-free method of Section 3, requires an available KB of seed facts that have supporting evidence in the corpus. Unfortunately, available KBs have low overlap with corpora in many languages, since KBs have cultural and geographical biases.

In response, we jointly model relation extraction in a high-resource language, such as English, and an alternative language with no such annotation available. The approach provides transfer learning of a predictive model to the alternative language, and generalizes naturally to modeling more languages.

Extending the training technique of Section 3 to corpora in multiple languages can be achieved by factorizing a matrix that mixes data from KB and from the two corpora. In Figure 1 we split the entities of a multilingual training corpus into sets depending on whether they have annotation in a KB and what corpora they appear in. We can perform transfer learning of a relation extractor to the low-resource language if there are entity pairs occurring in the two corpora, even if there is no KB annotation for these pairs. Note that we do not use the entity pair embeddings at test time: They simply used to bridge the languages during training. To form predictions in the low-resource language, we can simply apply the pattern scoring approach of Section 3.

In Section 6.2, we demonstrate that jointly learning models for English and Spanish, with no annotation for the Spanish data, provides fairly accurate Spanish AKBC, and even improves the performance of the English model. Note that we are not performing *zero-shot* learning of a Spanish relation extraction model (**?**). The relations in the target schema are language-independent concepts, and we have supervision for these in English.

5.1 Tied Sentence Encoders

The sentence encoder approach of Section 4 is complementary to our multilingual modeling technique: we simply use a separate encoder for each language. This approach is sub-optimal, however, because each sentence encoder will have a separate matrix of word embeddings for its vocabulary, despite the fact that there may be considerable shared structure between the languages. In re-

sponse, we propose a simple method for tying the parameters of the sentence encoders across languages.

Most work on multilingual word embeddings uses aligned sentences from the Europarl dataset (**?**) to align word embeddings across languages (**???**). Others (**??**) align separate single-language embedding models using a word-level alignment dictionary. Notably, **?** use translation pairs to learn a linear transform from one embedding space to another.

Drawing on these dictionary-based techniques, we first obtain a list of word-word translation pairs between the languages using a translation dictionary. The first layer of our deep text encoder consists of a word embedding lookup table. For the aligned word types, we use a single cross-lingual embedding. Details for our approach are described in Section A.4.

6 Experiments

6.1 Task

Much of the related work on embedding knowledge bases evaluates on the FB15k dataset (**?????**). Here, relation extraction is posed as link prediction on a subset of Freebase. This task does not capture the particular difficulties we address in this work: (1) evaluation on entities and text unseen during training, and (2) zero-annotation learning of a predictor for a low-resource language.

Instead, we focus on the 2013 TAC KBP slot-filling task. The aim of the TAC benchmark is to improve both coverage and quality of relation extraction evaluation compared to just checking the extracted facts against a knowledge base, which can be incomplete and where the provenances are not verified. In the slot-filling task, each system is given a set of paired query entities and relations or 'slots' to fill, and the goal is to correctly fill as many slots as possible along with provenance from the corpus. For example, given the query entity/relation pair (*Barack Obama, per:spouse*), the system should return the entity *Michelle Obama* along with sentence(s) whose text expresses that relation. The answers returned by all participating teams, along with a human search (with timeout), are judged manually for correctness, i.e. whether the provenance specified by the system indeed expresses the relation in question.

The state of the art systems on this task all rely on matching of handwritten patterns to find additional answers, while our models use only indirect supervision via entity pairs; even our AN heuristics are automatically generated. RelationFactory (**?**), the top-ranking system of the 2013 English slot-filling task, reports a score of 40.17. The highest F1 score on the 2013 slot-filling task is 40.86 (**?**) for a model that uses additional active learning annotation (the scores for this model were reported setting the optimal prediction thresholds on the 2013 set itself).

Our retrieval pipeline works by first generating all valid slot filler candidates for each query entity and slot, based on entities extracted from the corpus using FACTORIE (**?**) to perform tokenization, segmentation, and entity extraction. An entity pair qualifies as a candidate prediction if it meets the type criteria for the slot.[1] The TAC 2013 English and Spanish newswire corpora each contain about 1 million newswire documents from 2009–2012. The document retrieval and entity matching components of our relation extraction pipeline are based on RelationFactory (**?**), the top-ranked system of the 2013 English slot-filling task. We also use the English distantly supervised training data from this system, which aligns the TAC 2012 corpus to Freebase, and heuristically link all entity mentions from our text corpora to a Freebase entity using anchor text in Wikipedia. More details on alignment and linking are described in Appendix A.3.

We evaluate our Spanish models on the 2012 TAC Spanish slot-filling evaluation. Because this TAC track was never officially run, the coverage of facts in the available annotation is very small, resulting in many correct predictions being marked incorrectly as precision errors. In response, we manually annotated all results returned by the five models considered in Table 5. Precision and recall are calculated with respect to the union of the TAC annotation and our new labeling[2].

As discussed in Section 4.1, models using a deep sentence encoder and using a pattern lookup table have complementary strengths and weaknesses. In response, we also present results where we ensemble the outputs of the two models. We manually shift the models' thresholds to be more precision-biased, and take the union of the predictions returned by the two models. In contrast, **?**, add the confidence scores of the systems and then apply a threshold. We found that this ensembling approach does not adequately account for the qualitative distinction in types of prediction that each technique can make accurately.

Finally, note both **?** and **?** explore the pros and cons of learning embeddings for entity pairs vs. separate embeddings for each entity. As this is orthogonal to our contributions, we only consider entity pair embeddings, which performed best in both works.

[1] Due to the difficulty of retrieval and entity detection the maximum recall for relation predictions is limited. For this reason, **?** restrict the evaluation to answer candidates returned by their system and effectively rescaling recall. We do not perform such a re-scaling in our English results in order to compare to other reported results. Our Spanish numbers are rescaled. All scores reflect the 'anydoc' (relaxed) scoring to mitigate penalizing effects for systems not included in the evaluation pool.

[2] Following **?** we remove facts about undiscovered entities to correct for recall.

Table 2: Precision, recall and F1 of English-only models on the English TAC 2013 slot-filling task. LSTM+USchema ensemble outperforms any single model.

Model	Recall	Precision	F1
CNN	31.6	36.8	34.1
LSTM	**32.2**	39.6	35.5
USchema	29.4	**42.6**	34.8
USchema+LSTM	34.4	41.9	37.7
USchema+LSTM+AN	36.7	43.1	**39.7**
USchema+LSTM+AN+ES	—	—	**40.9**
?*	—	—	40.2
?*	—	—	40.9

6.2 Results

See Section A.6 for a discussion of the hyper-parameters, optimization techniques, etc. used in all experiments. As in **?**, we train using the BPR loss of **?**.

[[todo: say that we tuned these on sample training queries]]

Table 3: Zero-Annotation transfer learning F1 scores on 2012 Spanish TAC KBP slot-filling task. Adding a translation dictionary improves all encoder-based models. Ensembling LSTM and USchema models performs the best.

Model	Es+En	Es+En+dict
CNN	11.4	13.8
LSTM	10.7	15.2
USchema	16.3	—
USchema+LSTM	17.3	**20.0**

Table 4 presents the performance of our English models. First, observe that the LSTM substantially outperforms a CNN. Second, note that the LSTM achieves higher recall than USchema whereas USchema is more precision-biased. This confirms our hypothesis in Section 4.1 about the strengths and weaknesses of the two approaches. Unsurprisingly, ensembling the LSTM and USchema improves F1 by nearly 2 points over the strongest single model, USchema. Adding the alternative names (AN) technique described in Section A.3 increases F1 by an additional 2 points, resulting in an F1 score that is competitive with the state-of-the-art.

In Table 3, we analyze the effect of jointly learning English and Spanish models on English slot filling performance. Adding Spanish data improves scores of USchema and CNN, though the LSTM remains unaffected. Further tying the parameters of English and Spanish data by adding a translation dictionary further improves the CNN, and greatly improves the ensemble of USchema and LSTM, leading to 1.5 point increase in F1 over the ensemble of models trained on English alone. The boost in score resulting from dictionary typing suggests that with dictionary-tied parameters the LSTM can better leverage the Spanish data to find good relations that USchema is unable to find with only parameter tying through entities. Since USchema embeds entire OpenIE patterns, and not single words, parameters cannot be tied at the word level and so dictionary-tied results are not applicable to this model. The final rows shows that the alternate names heuristic is complementary to improvements from including Spanish.

Table 5 presents results for our Spanish relation extractors trained using zero-annotation transfer learning. For both the CNN and LSTM, tying word embeddings between the two languages results in substantial improvements. We see that ensembling the non-dictionary LSTM with USchema leads to a lower score than just USchema alone, but ensembling the dictionary-tied LSTM with USchema provides a significant increase of nearly 4 F1 points over the highest-scoring single model, USchema. Clearly, grounding the Spanish data using a translation dictionary provides much better Spanish word representations. These improvements are complementary to the baseline USchema model, and yield impressive results when ensembled.

6.3 Qualitative Analysis

Analysis of our English models suggests that our encoder-based models (LSTM) extract relations based on a wide range of semantically similar patterns that the pattern-matching model (USchema) is unable to score due to a lack of exact string match in the test data. For example, Table 6 lists three examples of the *per:children* relation that the LSTM finds which USchema does not, as well as three patterns that USchema does find. Though the LSTM patterns are all semantically and syntactically similar, they each contain different specific noun phrases, e.g. *Lori*, *four children*, *toddler daughter*, *Lee and Albert*, etc. Because these specific nouns weren't seen during training, USchema fails to find these patterns whereas the LSTM learns to ignore the specific nouns in favor of the overall pattern, that of a parent-child relationship in an obituary. USchema is limited to finding the relations represented by patterns observed during training, which limits the patterns matched at test-time to short and common patterns; all the USchema patterns matched at test time were similar to those listed in Table 6: variants of *'s son, '*.

Analysis of our multilingual models also suggests that they successfully embed semantically similar relations across languages using tied entity pairs and translation dictionary as grounding. Table 7 lists three top nearest neighbors in English for several Spanish patterns from the text. In each case, the English patterns capture the relation represented in the Spanish text.

Table 4: Examples of the *per:children* relation discovered by the LSTM and Universal Schema. Entities are bold and patterns italicized. The LSTM can model a richer set of patterns

LSTM
McGregor *is survived by his wife, Lori, and four children, daughters Jordan,* **Taylor** and Landri, and a son, Logan.
In addition to his wife, **Mays** *is survived by a toddler daughter and a son,* **Billy Mays Jr.**, who is in his 20s.
Anderson *is survived by his wife Carol, sons Lee and Albert, daughter* **Shirley Englebrecht** and nine grandchildren.
USchema
Dio *'s son,* **Dan Padavona**, cautioned the memorial crowd to be screened regularly by a doctor and take care of themselves, something he said his father did not do.
But **Marshall** *'s son,* **Philip**, told a different story.
"I'd rather have Sully doing this than some stranger, or some hotshot trying to be the next Billy Mays," said the guy who actually is the next **Billy Mays***, his son* **Billy Mays III**.

Table 5: Top English patterns for a Spanish query pattern encoded using the dictionary LSTM: For each Spanish query (English translation in italics), a list of English nearest neighbors.

arg1 y cuatro de sus familias, incluidos su esposa, Wu Shu-chen, su hijo, *arg2* arg1 *and four of his family members, including his wife, Wu Shu-chen, his son,* arg2
arg1 and his son *arg2* *arg1* is survived by his wife, Sybil MacKenzie and a son, *arg2* *arg1* gave birth to a baby last week – son *arg2*
arg1 (Puff Daddy, cuyos verdaderos nombre sea *arg2* arg1 *(Puff Daddy, whose real name is* arg2
arg1 (usually credited as *E1* *arg1* (also known as Gero ##, real name *arg2* *arg1* and (after changing his name to *arg2*
arg1 llegó a la alfombra roja en compañía de su esposa, la actriz Suzy Amis, casi una hora antes que su ex esposa, *arg2* arg1 *arrived on the red carpet with his wife, actress Suzy Amis, nearly an hour before his ex-wife ,* arg2
arg1, who may or may not be having twins with husband *arg2* *arg1*, aged twenty, Kirk married *arg2* *arg1* went to elaborate lengths to keep his wedding to former supermodel *arg2*

In addition to embedding semantically similar phrases from English and Spanish to have high similarity, our models also learn high-quality multilingual word embeddings. In Table 8 we compare Spanish nearest neighbors of English query words learned by the LSTM with dictionary ties versus the LSTM with no ties, using no unsupervised pre-training for the embeddings. Both approaches jointly embed Spanish and English word types, using shared entity embeddings, but the dictionary-tied model learns qualitatively better multilingual embeddings.

7 Conclusion

By jointly embedding English and Spanish KBs, we can train an accurate Spanish relation extraction model using no direct annotation for relations in the Spanish data. This approach has the added benefit of providing significant accuracy improvements for the English model, obtaining nearly state-of-the-art accuracy on the 2013 TAC KBC slot filling task, while using substantially fewer hand-coded rules than alternative systems. By using deep sentence encoders, we can perform prediction for arbitrary input text and for entities unseen in training. Sentence encoders also provides opportunities to improve cross-lingual transfer learning by sharing word embeddings across languages. In future work we will apply this model to many more languages and domains besides newswire text. We would also like to avoid the entity detection problem by using a deep architecture to both identify entity mentions and identify relations between them.

Acknowledgments

Many thanks to Arvind Neelakantan and Noah Smith for good ideas and discussions. We also appreciate a generous hardware grant from nVidia. This work was supported in part by the Center for Intelligent Information Retrieval, in part by Defense Advanced Research Projects Agency (DARPA) under agreement #FA8750-13-2-0020 and contract #HR0011-15-2-0036, and in part by the National Science Foundation (NSF) grant numbers DMR-1534431, IIS-1514053 and CNS-0958392. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon, in part by DARPA via agreement #DFA8750-13-2-0020 and NSF grant #CNS-0958392. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

A Appendix

A.1 Additional Qualitative Results

Our model jointly embeds KB relations together with English and Spanish text. We demonstrate that plausible textual patterns are embedded close to the KB relations they express. Table 9 shows top scoring English and Spanish patterns given sample relations from our TAC KB.

A.2 Details Concerning Cosine Similarity Computation

We measure the similarity between r_{text} and r_{schema} by computing the vectors' cosine similarity. However, such a distance is not well-defined, since the model was trained

Table 6: Example English query words (not in translation dictionary) in bold with their top nearest neighbors by cosine similarity listed for the dictionary and no ties LSTM variants. Dictionary-tied nearest neighbors are consistently more relevant to the query word than untied.

CEO	
Dictionary	No Ties
jefe (chief)	CEO
CEO	director (principle)
ejecutivo (executive)	directora (director)
cofundador (cofounder)	firma (firm)
president (chairman)	magnate (tycoon)
headquartered	
Dictionary	No Ties
sede (headquarters)	Geológico (Geological)
situado (located)	Treki (Treki)
selectivo (selective)	Geofísico(geophysical)
profesional (vocational)	Normandía (Normandy)
basándose (based)	emplea (uses)
hubby	
Dictionary	No Ties
matrimonio (marriage)	esposa (wife)
casada (married)	esposo (husband)
esposa (wife)	casada(married)
casó (married)	embarazada (pregnant)
embarazada (pregnant)	embarazo (pregnancy)
alias	
Dictionary	No Ties
simplificado (simplified)	Weaver (Weaver)
sabido (known)	interrogación (question)
seudónimo (pseudonym)	alias
privatización (privatisation)	reelecto (reelected)
nombre (name)	conocido (known)

Table 7: Top scoring patterns for both Spanish (top) and English (bottom) given query TAC relations.

per:sibling
arg1, según petición the primeros ministro, su hermano gemelo *arg2*
arg1, sea the principal favorito para esto oficina que también ambiciona su hermano *arg2*
arg1, y su hermano gemelo, the primeros ministro *arg2*
arg1, for whose brother *arg2*
arg1 inherited his brother *arg2*
arg1 on saxophone and brother *arg2*
org:top_members_employees
arg2, presidente y director generales the *arg1*
arg2, presidente of the negocios especializada *arg1*
arg2 (CIA), the director of the entidad, *arg1*
arg2, vice president and policy director of the *arg1*
arg2, president of the German Soccer *arg1*
arg2, president of the quasi-official *arg1*
per:alternate_names
arg1 (como también son sabido para *arg2*
arg2-cuyos verdaderos nombre sea *arg1*
arg1 también sabido como *arg2*
arg1 aka *arg2*
arg1, who also creates music under the pseudonym *arg2*
arg1 (of Modern Talking fame) aka *arg2*
per:cities_of_residence
arg1, poblado dónde vive *arg2*
arg1, una ciudadano naturalizado american y nacido in *arg2*
arg1, que vive in *arg2*
arg1 was born Jan. # , #### in *arg2*
arg1 was born on Monday in *arg2*
arg1 was born at Keighley in *arg2*

using inner products between entity vectors and relation vectors, not between two relation vectors. The US likelihood is invariant to invertible transformations of the latent coordinate system, since $\sigma\left(u_{s,o}^\top v_r\right) = \sigma\left((A^\top u_{s,o})^\top A^{-1} v_r\right)$ for any invertible A. When taking inner products between two v terms, however, the implicit A^{-1} terms do not cancel out. We found that this issue can be minimized, and high quality predictive accuracy can be achieved, simply by using sufficient ℓ_2 regularization to avoid implicitly learning an A that substantially stretches the space.

A.3 Data Pre-processing, Distant Supervision and Extraction Pipeline

We replace tokens occurring less than 5 times in the corpus with UNK and normalize all digits to # (e.g. Oct-11-1988 becomes Oct-##-####). For each sentence, we then extract all entity pairs and the text between them as surface patterns, ignoring patterns longer than 15 tokens. This results in 48 million English 'relations'. In Section A.5, we describe a technique for normalizing the surface patterns. We filter out entity pairs that occurred less than 10 times in the data and extract the largest connected component in this entity co-occurrence graph. This is necessary for the baseline US model, as otherwise learning decouples into independent problems per connected component. Though the components are connected when using sentence encoders, we use only a single component to facilitate a fair comparison between modeling approaches. We add the distant supervision training facts from the RelationFactory system, i.e. 352,236 entity-pair-relation tuples obtained from Freebase and high precision seed patterns. The final training data contains a set of 3,980,164 (KB and openIE) facts made up of 549,760 unique entity pairs, 1,285,258 unique relations and 62,841 unique tokens. For entity linking, we make use of the fact that most Freebase entries contain a link to the corresponding Wikipedia page, and we heuristically link all entity mentions from our text corpora to a

Freebase entity by the following process: First, a set of candidate entities is obtained by following frequent link anchor text statistics. We then select that candidate entity for which the cosine similarity between the respective Wikipedia and the sentence context of the mention is highest, and link to that entity if a threshold is exceeded.

We perform the same preprocessing on the Spanish data, resulting in 34 million raw surface patterns between entities. We then filter patterns that never occur with an entity pair found in the English data. This yields 860,502 Spanish patterns. Our multilingual model is trained on a combination of these Spanish patterns, the English surface patterns, and the distant supervision data described above. We learn word embeddings for 39,912 unique Spanish word types. After parameter tying for translation pairs (Section 5.1), there are 33,711 additional Spanish words not tied to English.

We also report results including an alternate names (AN) heuristic, which uses automatically-extracted rules to detect the 'alternate name' relation. For this, frequent Wikipedia link anchor texts are collected for each query entity. If a high probability anchor text co-occurs with the canonical name of the query in the same document, we return the anchor text as a slot fill.

A.4 Generation of Cross-Lingual Tied Word Types

We follow the same procedure for generating translation pairs as **?**. First, we select the top 6000 words occurring in the lowercased Europarl dataset for each language and obtain a Google translation. We then filter duplicates and translations resulting in multi-word phrases. We also remove English past participles (ending in -ed) as we found the Google translation interprets these as adjectives (e.g., 'she read the borrowed book' rather than 'she borrowed the book') and much of the relational structure in language we seek to model is captured by verbs. This resulted in 6201 translation pairs that occurred in our text corpus. Though higher quality translation dictionaries would likely improve this technique, our experimental results show that such automatically generated dictionaries perform well.

A.5 Open IE Pattern Normalization

To improve US generalization, our US relations use log-shortened patterns where the middle tokens in patterns longer than five tokens are simplified. For each long pattern we take the first two tokens and last two tokens, and replace all k remaining tokens with the number $\log k$. For example, the pattern **Barack Obama** *is married to a person named* **Michelle Obama** would be converted to: **Barack Obama** *is married [1] person named* **Michell Obama**. This shortening performs slightly better than whole patterns. LSTM and CNN variants use the entire sequence of tokens.

A.6 Implementation and Hyperparameters

All models are implemented in Torch[3] and tuned to maximize F1 on the TAC 2012 slot-filling evaluation. We additionally tune the thresholds of our pattern scorer on a per-relation basis to maximize F1 using the 2012 TAC KBP slot filling evaluation as a validation set. All experiments use 50-dimensional relation and entity pair embeddings. Our CNN is implemented as described in **?**, using width-3 convolutions, followed by tanh and max pool layers. The CNN and LSTM both learned 100-dimensional word embeddings, which were randomly initialized. We found that pre-trained word embeddings did not substantially affect the results. Entity pair embeddings for the baseline US model are randomly initialized. For the models with LSTM and CNN text encoders, entity pair embeddings are initialized using vectors from the baseline US model. This performs better than random initialization. We perform ℓ_2 gradient clipping to 1 on all models. Universal Schema uses a batch size of 1024 while the CNN and LSTM use 128. All models are optimized using ADAM (**?**) with $\epsilon = 1e-8$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ with a learning rate of .001 for US and .0001 for CNN and LSTM. The CNN and LSTM also use dropout of 0.1 after the embedding layer. All models are trained for a maximum of 15 epochs. We also experimented with bidirectional LSTMs which did not perform better.

[3] http://torch.ch