Position-aware Attention and Supervised Data Improve Slot Filling

Yuhao Zhang, Victor Zhong, Danqi Chen,
Gabor Angeli, Christopher D. Manning
Stanford University
Stanford, CA 94305
{yuhao, vzhong, danqi}@cs.stanford.edu
{angeli, manning}@cs.stanford.edu

Abstract

Organized relational knowledge in the form of "knowledge graphs" is important for many applications. However, the ability to populate knowledge bases with facts automatically extracted from documents has improved frustratingly slowly. This paper simultaneously addresses two issues that have held back prior work. We first propose an effective new model, which combines an LSTM sequence model with a form of entity position-aware attention that is better suited to relation extraction. Then we build TACRED, a large (119,474 examples) supervised relation extraction dataset, obtained via crowdsourcing and targeted towards TAC KBP relations. The combination of better supervised data and a more appropriate high-capacity model enables much better relation extraction performance. When the model trained on this new dataset replaces the previous relation extraction component of the best TAC KBP 2015 slot filling system, its F_1 score increases markedly from 22.2% to 26.7%.

1 Introduction

A basic but highly important challenge in natural language understanding is being able to populate a knowledge base with relational facts contained in a piece of text. For the text shown in Figure 1, the system should extract triples, or equivalently, knowledge graph edges, such as ⟨*Penner*, *per:spouse*, *Lisa Dillman*⟩. Combining such extractions, a system can produce a knowledge graph of relational facts between persons, organizations, and locations in the text. This task involves entity recognition, mention coreference and/or entity linking, and relation extraction; we focus on the most challenging "slot filling" task of filling in the relations between entities in the text.

Penner *is survived by his brother,* ***John****, a* ***copy editor*** *at the* ***Times****, and his former wife,* ***Times sportswriter Lisa Dillman****.*

Subject	Relation	Object
Mike Penner	per:spouse	Lisa Dillman
Mike Penner	per:siblings	John Penner
Lisa Dillman	per:title	Sportswriter
Lisa Dillman	per:employee_of	Los Angeles Times
John Penner	per:title	Copy Editor
John Penner	per:employee_of	Los Angeles Times

Figure 1: An example of relation extraction from the TAC KBP corpus.

Organized relational knowledge in the form of "knowledge graphs" has become an important knowledge resource. These graphs are now extensively used by search engine companies, both to provide information to end-users and internally to the system, as a way to understand relationships. However, up until now, automatic knowledge extraction has proven sufficiently difficult that most of the facts in these knowledge graphs have been built up by hand. It is therefore a key challenge to show that NLP technology can effectively contribute to this important problem.

Existing work on relation extraction (e.g., Zelenko et al., 2003; Mintz et al., 2009; Adel et al., 2016) has been unable to achieve sufficient recall or precision for the results to be usable versus hand-constructed knowledge bases. Supervised training data has been scarce and, while techniques like distant supervision appear to be a promising way to extend knowledge bases at low cost, in practice the training data has often been too noisy for reliable training of relation extraction systems (Angeli et al., 2015). As a result most systems fail to make correct extractions even in apparently straightforward cases like Figure 1,

Example	Entity Types & Label
Carey will succeed **Cathleen P. Black**, who held the position for 15 years and will take on a new role as **chairwoman** of Hearst Magazines, the company said.	**Types:** PERSON/TITLE **Relation:** *per:title*
Irene Morgan Kirkaldy, who was born and reared in **Baltimore**, lived on Long Island and ran a child-care center in Queens with her second husband, Stanley Kirkaldy.	**Types:** PERSON/CITY **Relation:** *per:city_of_birth*
Pandit worked at the brokerage Morgan Stanley for about 11 years until 2005, when he and some Morgan Stanley colleagues quit and later founded the hedge fund **Old Lane Partners**.	**Types:** ORGANIZATION/PERSON **Relation:** *org:founded_by*
Baldwin declined further comment, and said JetBlue chief **executive** Dave Barger was unavailable.	**Types:** PERSON/TITLE **Relation:** *no_relation*

Table 1: Sampled examples from the TACRED dataset. Subject entities are highlighted in blue and object entities are highlighted in red.

where the best system at the NIST TAC Knowledge Base Population (TAC KBP) 2015 evaluation failed to recognize the relation between *Penner* and *Dillman*.[1] Consequently most automatic systems continue to make heavy use of hand-written rules or patterns because it has been hard for machine learning systems to achieve adequate precision or to generalize as well across text types. We believe machine learning approaches have suffered from two key problems: (1) the models used have been insufficiently tailored to relation extraction, and (2) there has been insufficient annotated data available to satisfy the training of data-hungry models, such as deep learning models.

This work addresses both of these problems. We propose a new, effective neural network sequence model for relation classification. Its architecture is better customized for the slot filling task: the word representations are augmented by extra distributed representations of word position relative to the subject and object of the putative relation. This means that the neural attention model can effectively exploit the combination of semantic similarity-based attention and position-based attention. Secondly, we markedly improve the availability of supervised training data by using Mechanical Turk crowd annotation to produce a large supervised training dataset (Table 1), suitable for the common relations between people, organizations and locations which are used in the TAC KBP evaluations. We name this dataset the **TAC R**elation **E**xtraction **D**ataset (TACRED), and will make it available through the Linguistic Data Consortium (LDC) in order to respect copyrights on the underlying text.

Combining these two gives a system with markedly better slot filling performance. This is shown not only for a relation classification task on the crowd-annotated data but also for the incorporation of the resulting classifiers into a complete cold start knowledge base population system. On TACRED, our system achieves a relation classification F_1 score that is 7.9% higher than that of a strong feature-based classifier, and 3.5% higher than that of the best previous neural architecture that we re-implemented. When this model is used in concert with a pattern-based system on the TAC KBP 2015 Cold Start Slot Filling evaluation data, the system achieves an F_1 score of 26.7%, which exceeds the previous state-of-the-art by 4.5% absolute. While this performance certainly does not solve the knowledge base population problem – achieving sufficient recall remains a formidable challenge – this is nevertheless notable progress.

2 A Position-aware Neural Sequence Model Suitable for Relation Extraction

Existing work on neural relation extraction (e.g., Zeng et al., 2014; Nguyen and Grishman, 2015; Zhou et al., 2016) has focused on convolutional neural networks (CNNs), recurrent neural networks (RNNs), or their combination. While these models generally work well on the datasets they are tested on, as we will show, they often fail to generalize to the longer sentences that are common in real-world text (such as in TAC KBP).

We believe that existing model architectures suffer from two problems: (1) Although modern sequence models such as Long Short-Term Memory (LSTM) networks have gating mechanisms to control the relative influence of each individual word to the final sentence representation (Hochreiter and Schmidhuber, 1997), these controls are not explicitly conditioned on the entire sentence being classified; (2) Most existing work either

[1] Note: former spouses count as spouses in the ontology.

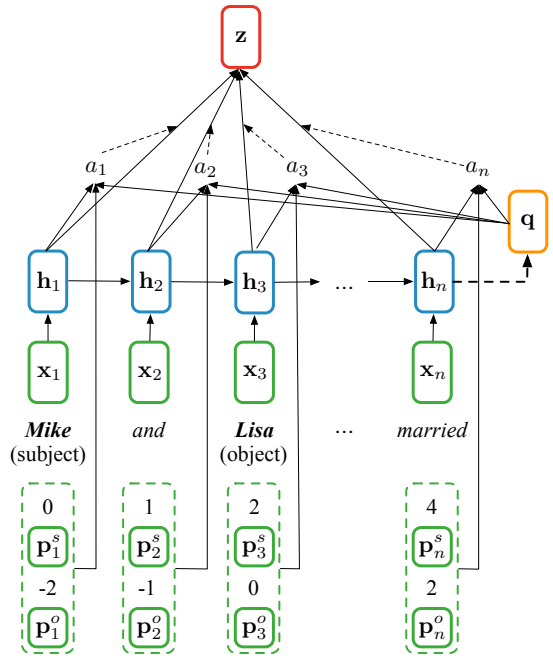

Figure 2: Our proposed position-aware neural sequence model. The model is shown with an example sentence *Mike and Lisa got married*.

does not explicitly model the positions of entities (i.e., subject and object) in the sequence, or models the positions only within a local region.

Here, we propose a new neural sequence model with a **position-aware attention mechanism** over an LSTM network to tackle these challenges. This model can (1) evaluate the relative contribution of each word after seeing the entire sequence, and (2) base this evaluation not only on the semantic information of the sequence, but also on the global positions of the entities within the sequence.

We formalize the relation extraction task as follows: Let $\mathcal{X} = [x_1, ..., x_n]$ denote a sentence, where x_i is the i-th token. A subject entity s and an object entity o are identified in the sentence, corresponding to two non-overlapping consecutive spans: $\mathcal{X}_s = [x_{s_1}, x_{s_1+1}, \ldots, x_{s_2}]$ and $\mathcal{X}_o = [x_{o_1}, x_{o_1+1}, \ldots, x_{o_2}]$. Given the sentence $\mathcal{X}$ and the positions of s and o, the goal is to predict a relation $r \in \mathcal{R}$ ($\mathcal{R}$ is the set of relations) that holds between s and o or no relation otherwise.

Inspired by the position encoding vectors used in Collobert et al. (2011) and Zeng et al. (2014), we define a position sequence relative to the subject entity $[p_1^s, ..., p_n^s]$, where

$$p_i^s = \begin{cases} i - s_1, & i < s_1 \\ 0, & s_1 \le i \le s_2 \\ i - s_2, & i > s_2 \end{cases} \quad (1)$$

Here s_1, s_2 are the starting and ending indices of the subject entity respectively, and $p_i^s \in \mathbb{Z}$ can be viewed as the relative distance of token x_i to the subject entity. Similarly, we obtain a position sequence $[p_1^o, ..., p_n^o]$ relative to the object entities.

Let $\mathbf{x} = [\mathbf{x}_1, ..., \mathbf{x}_n]$ be word embeddings of the sentence, obtained using an embedding matrix $\mathbf{E}$. Similarly, we obtain position embedding vectors $\mathbf{p}^s = [\mathbf{p}_1^s, ..., \mathbf{p}_n^s]$ and $\mathbf{p}^o = [\mathbf{p}_1^o, ..., \mathbf{p}_n^o]$ using a shared position embedding matrix $\mathbf{P}$ respectively. Next, as shown in Figure 2, we obtain hidden state representations of the sentence by feeding $\mathbf{x}$ into an LSTM:

$$\{\mathbf{h}_1, ..., \mathbf{h}_n\} = \text{LSTM}(\{\mathbf{x}_1, ..., \mathbf{x}_n\}) \quad (2)$$

We define a *summary* vector $\mathbf{q} = \mathbf{h}_n$ (i.e., the output state of the LSTM). This summary vector encodes information about the entire sentence. Then for each hidden state $\mathbf{h}_i$, we calculate an attention weight a_i as:

$$u_i = \mathbf{v}^\top \tanh(\mathbf{W}_h \mathbf{h}_i + \mathbf{W}_q \mathbf{q} + \mathbf{W}_s \mathbf{p}_i^s + \mathbf{W}_o \mathbf{p}_i^o) \quad (3)$$

$$a_i = \frac{\exp(u_i)}{\sum_{j=1}^{n} \exp(u_j)} \quad (4)$$

Here $\mathbf{W}_h, \mathbf{W}_q \in \mathbb{R}^{d_a \times d}$, $\mathbf{W}_s, \mathbf{W}_o \in \mathbb{R}^{d_a \times d_p}$ and $\mathbf{v} \in \mathbb{R}^{d_a}$ are learnable parameters of the network, where d is the dimension of hidden states, d_p is the dimension of position embeddings, and d_a is the size of attention layer. Additional parameters of the network include embedding matrices $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times d}$ and $\mathbf{P} \in \mathbb{R}^{(2L-1) \times d_p}$, where $\mathcal{V}$ is the vocabulary and L is the maximum sentence length.

We regard attention weight a_i as the relative contribution of the specific word to the sentence representation. The final sentence representation $\mathbf{z}$ is computed as:

$$\mathbf{z} = \sum_{i=1}^{n} a_i \mathbf{h}_i \quad (5)$$

$\mathbf{z}$ is later fed into a fully-connected layer followed by a softmax layer for relation classification.

Note that our model significantly differs from the attention mechanism in Bahdanau et al. (2015) and Zhou et al. (2016) in our use of the summary vector and position embeddings, and the way our attention weights are computed. An intuitive way to understand the model is to view the attention calculation as a selection process, where the goal is to select relevant contexts over irrelevant ones.

Dataset	# Rel.	# Ex.	% Neg.
SemEval-2010 Task 8	19	10,717	17.4%
ACE 2003–2004	24	16,771	N/A
TACRED	42	119,474	78.7%

Table 2: A comparison of existing datasets and our proposed TACRED dataset. % Neg. denotes the percentage of negative examples (no relation).

Here the summary vector ($\mathbf{q}$) helps the model to base this selection on the semantic information of the entire sentence (rather than on each word only), while the position vectors ($\mathbf{p}_i^s$ and $\mathbf{p}_i^o$) provides important spatial information between each word and the entities.

3 The TAC Relation Extraction Dataset

Previous research has shown that slot filling systems can greatly benefit from supervised data. For example, Angeli et al. (2014b) showed that even a small amount of supervised data can boost the end-to-end F_1 score by 3.9% on the TAC KBP tasks. However, existing relation extraction datasets such as the SemEval-2010 Task 8 dataset (Hendrickx et al., 2009) and the Automatic Content Extraction (ACE) (Strassel et al., 2008) dataset are less useful for this purpose. This is mainly because: (1) these datasets are relatively small for effectively training high-capacity models (see Table 2), and (2) they capture very different types of relations. For example, the SemEval dataset focuses on semantic relations (e.g., *Cause-Effect*, *Component-Whole*) between two nominals.

One can further argue that it is easy to obtain a large amount of training data using distant supervision (Mintz et al., 2009). In practice, however, due to the large amount of noise in the induced data, training relation extractors that perform well becomes very difficult. For example, Riedel et al. (2010) show that up to 31% of the distantly supervised labels are wrong when creating training data from aligning Freebase to newswire text.

To tackle these challenges, we collect a large supervised dataset TACRED, targeted towards the TAC KBP relations.

Data collection. We create TACRED based on query entities and annotated system responses in the yearly TAC KBP evaluations. In each year of the TAC KBP evaluation (2009–2015), 100 entities (people or organizations) are given as queries, for which participating systems should find associated relations and object entities. We make use of Mechanical Turk to annotate each sentence in the source corpus that contains one of these query entities. For each sentence, we ask crowd workers to annotate both the subject and object entity spans and the relation types.

Data Split	# Ex.	Years
Train	75,050	2009–2012
Dev	25,764	2013
Test	18,660	2014

Table 3: Statistics on TACRED: number of examples and the source of each portion.

Dataset stratification. In total we collect 119,474 examples. We stratify TACRED across years in which the TAC KBP challenge was run, and use examples corresponding to query entities from 2009 to 2012 as training split, 2013 as development split, and 2014 as test split. We reserve the TAC KBP 2015 evaluation data for running slot filling evaluations, as presented in Section 4. Detailed statistics are given in Table 3.

Discussion. Table 1 presents sampled examples from TACRED. Compared to existing datasets, TACRED has four advantages. First, it contains an order of magnitude more relation instances (Table 2), enabling the training of expressive models. Second, we reuse the entity and relation types of the TAC KBP tasks. We believe these relation types are of more interest to downstream applications. Third, we fully annotate all negative instances that appear in our data collection process, to ensure that models trained on TACRED are not biased towards predicting false positives on real-world text. Lastly, the average sentence length in TACRED is 36.2, compared to 19.1 in the SemEval dataset, reflecting the complexity of contexts in which relations occur in real-world text.

Due to space constraints, we describe the data collection and validation process, system interfaces, and more statistics and examples of TACRED in the supplementary material. We will make TACRED publicly available through the LDC.

4 Experiments

In this section we evaluate the effectiveness of our proposed model and TACRED on improving slot

filling systems. Specifically, we run two sets of experiments: (1) we evaluate model performance on the relation extraction task using TACRED, and (2) we evaluate model performance on the TAC KBP 2015 cold start slot filling task, by training the models on TACRED.

4.1 Baseline Models

We compare our model against the following baseline models for relation extraction and slot filling:

TAC KBP 2015 winning system. To judge our proposed model against a strong baseline, we compare against Stanford's top performing system on the TAC KBP 2015 cold start slot filling task (Angeli et al., 2015). At the core of this system are two relation extractors: a pattern-based extractor and a logistic regression (LR) classifier. The pattern-based system uses a total of 4,528 surface patterns and 169 dependency patterns. The logistic regression model was trained on approximately 2 million bootstrapped examples (using a small annotated dataset and high-precision pattern system output) that are carefully tuned for TAC KBP slot filling evaluation. It uses a comprehensive feature set similar to the MIML-RE system for relation extraction (Surdeanu et al., 2012), including lemmatized n-grams, sequence NER tags and POS tags, positions of entities, and various features over dependency paths, etc.

Convolutional neural networks. We follow the 1-dimensional CNN architecture by Nguyen and Grishman (2015) for relation extraction. This model learns a representation of the input sentence, by first running a series of convolutional operations on the sentence with various filters, and then feeding the output into a max-pooling layer to reduce the dimension. The resulting representation is then fed into a fully-connected layer followed by a softmax layer for relation classification. As an extension, positional embeddings are also introduced into this model to better capture the relative position of each word to the subject and object entities and were shown to achieve improved results. We use "CNN-PE" to represent the CNN model with positional embeddings.

Dependency-based recurrent neural networks. In dependency-based neural models, shortest dependency paths between entities are often used as input to the neural networks. The intuition is to eliminate tokens that are potentially less relevant to the classification of the relation. For the example in Figure 1, the shortest dependency path between the two entities is:

$$\text{[Penner]} \leftarrow \text{survived} \rightarrow \text{brother} \rightarrow \text{wife} \rightarrow \text{[Lisa Dillman]}$$

We follow the SDP-LSTM model proposed by Xu et al. (2015b). In this model, each shortest dependency path is divided into two separate sub-paths from the subject entity and the object entity to the lowest common ancestor node. Each sub-path is fed into an LSTM network, and the resulting hidden units at each word position are passed into a max-over-time pooling layer to form the output of this sub-path. Outputs from the two sub-paths are then concatenated to form the final representation.

In addition to the above models, we also compare our proposed model against an LSTM sequence model without attention mechanism.

4.2 Implementation Details

We map words that occur less than 2 times in the training set to a special $<$*UNK*$>$ token. We use the pre-trained GloVe vectors (Pennington et al., 2014) to initialize word embeddings. For all the LSTM layers, we find that 2-layer stacked LSTMs generally work better than one-layer LSTMs. We minimize cross-entropy loss over all 42 relations using *AdaGrad* (Duchi et al., 2011). We apply Dropout with $p = 0.5$ to CNNs and LSTMs. During training we also find a word dropout strategy to be very effective: we randomly set a token to be $<$*UNK*$>$ with a probability p. We set p to be 0.06 for the SDP-LSTM model and 0.04 for all other models.

Entity masking. We replace each subject entity in the original sentence with a special $<$*NER*$>$-*SUBJ* token where $<$*NER*$>$ is the corresponding NER signature of the subject as provided in TACRED. We do the same processing for object entities. This processing step helps (1) provide a model with entity type information, and (2) prevent a model from overfitting its predictions to specific entities.

Multi-channel augmentation. Instead of using only word vectors as input to the network, we augment the input with part-of-speech (POS) and named entity recognition (NER) embeddings. We run Stanford CoreNLP (Manning et al., 2014) to obtain the POS and NER annotations.

	Model	P	R	F_1
Traditional	Patterns	**85.3**	23.4	36.8
	LR	72.0	47.8	57.5
	LR + Patterns	71.4	50.1	58.9
Neural	CNN	72.1	50.3	59.2
	CNN-PE	68.2	55.4	61.1
	SDP-LSTM	62.0	54.8	58.2
	LSTM	61.4	61.7	61.5
	Our model	67.7	**63.2**	**65.4**
	Ensemble	69.4	**64.8**	**67.0**

Table 4: Model performance on the test set of TACRED, micro-averaged over instances. LR = Logistic Regression.

We describe our model hyperparameters and training in detail in the supplementary material.

4.3 Evaluation on TACRED

We first evaluate all models on TACRED. We train each model for 5 separate runs with independent random initializations. For each run we perform early stopping using the dev set. We then select the run (among 5) that achieves the *median* F_1 score on the dev set, and report its test set performance.

Table 4 summarizes our results. We observe that all neural models achieve higher F_1 scores than the logistic regression and patterns systems, which demonstrates the effectiveness of neural models for relation extraction. Although positional embeddings help increase the F_1 by around 2% over the plain CNN model, a simple (2-layer) LSTM model performs surprisingly better than CNN and dependency-based models. Lastly, our proposed position-aware mechanism is very effective and achieves an F_1 score of 65.4%, with an absolute increase of 3.9% over the best baseline neural model (LSTM) and 7.9% over the baseline logistic regression system. We also run an ensemble of our position-aware attention model which takes majority votes from 5 runs with random initializations and it further pushes the F_1 score up by 1.6%.

We find that different neural architectures show a different balance between precision and recall. CNN-based models tend to have higher precision; RNN-based models have better recall. This can be explained by noting that the filters in CNNs are essentially a form of "fuzzy n-gram patterns".

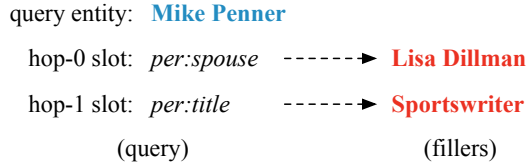

Figure 3: An example query and corresponding fillers in the TAC KBP cold start slot filling task.

4.4 Evaluation on TAC KBP Slot Filling

Second, we evaluate the slot filling performance of all models using the TAC KBP 2015 cold start slot filling task (Ellis et al., 2015). In this task, about 50k newswire and Web forum documents are selected as the evaluation corpus. A slot filling system is asked to answer a series of queries with two-hop slots (Figure 3): The first slot asks about fillers of a relation with the query entity as the subject (Mike Penner), and we term this a **hop-0** slot; the second slot asks about fillers with the system's hop-0 output as the subject, and we term this a **hop-1** slot. System predictions are then evaluated against gold annotations, and micro-averaged precision, recall and F_1 scores are calculated at the hop-0 and hop-1 levels. Lastly **hop-all** scores are calculated by combining hop-0 and hop-1 scores.[2]

Evaluating relation extraction systems on slot filling is particularly challenging in that: (1) End-to-end cold start slot filling scores conflate the performance of all modules in the system (i.e., entity recognizer, entity linker and relation extractor). (2) Errors in hop-0 predictions can easily propagate to hop-1 predictions. To fairly evaluate each relation extraction model on this task, we use Stanford's 2015 slot filling system as our basic pipeline.[3] It is a very strong baseline specifically tuned for TAC KBP evaluation and ranked top in the 2015 evaluation. We then plug in the corresponding relation extractor trained on TACRED, keeping all other modules unchanged.

Table 5 presents our results. We find that: (1) by only training our logistic regression model on TACRED (in contrast to on the 2 million bootstrapped examples used in the 2015 Stanford system) and combining it with patterns, we obtain a higher hop-0 F_1 score than the 2015 Stanford sys-

[2]In the TAC KBP cold start slot filling evaluation, a hop-1 slot is transferred to a pseudo-slot which is treated equally as a hop-0 slot. Hop-all precision, recall and F1 are then calculated by combining these pseudo-slot predictions and hop-0 predictions.

[3]This system uses the fine-grained NER system in Stanford CoreNLP (Manning et al., 2014) for entity detection and the Illinois Wikifier (Ratinov et al., 2011) for entity linking.

Model	Hop-0			Hop-1			Hop-all		
	P	R	F_1	P	R	F_1	P	R	F_1
Patterns	**63.8**	17.7	27.7	**49.3**	8.6	14.7	**58.9**	13.3	21.8
LR	36.6	21.9	27.4	15.1	10.1	12.2	25.6	16.3	19.9
+ Patterns (2015 winning system)	37.5	24.5	29.7	16.5	12.8	14.4	26.6	19.0	22.2
LR trained on TACRED	32.7	20.6	25.3	7.9	9.5	8.6	16.8	15.3	16.0
+ Patterns	36.5	26.5	30.7	11.0	15.3	12.8	20.1	21.2	20.6
Our model	39.0	28.9	33.2	17.7	13.9	15.6	28.2	21.5	24.4
+ Patterns	40.2	**31.5**	**35.3**	19.4	**16.5**	**17.8**	29.7	**24.2**	**26.7**

Table 5: Model performance on TAC KBP 2015 slot filling evaluation, micro-averaged over queries. Hop-0 scores are calculated on the simple single-hop slot filling results; hop-1 scores are calculated on slot filling results chained on systems' hop-0 predictions; hop-all scores are calculated based on the combination of the two. LR = logistic regression.

Model	Dev F_1
Final Model	**66.22**
– Position-aware attention	65.12
– Attention	64.71
– Pre-trained embeddings	65.34
– Word dropout	65.69
– All above	63.60

Table 6: An ablation test of our position-aware attention model, evaluated on TACRED dev set. Scores are median of 5 models.

tem, and a similar hop-all F_1; (2) our proposed position-aware attention model substantially outperforms the 2015 Stanford system on all hop-0, hop-1 and hop-all F_1 scores. Combining it with the patterns, we achieve a hop-all F_1 of 26.7%, an absolute improvement of 4.5% over the previous state-of-the-art result.

4.5 Analysis

Model ablation. Table 6 presents the results of an ablation test of our position-aware attention model on the development set of TACRED. The entire attention mechanism contributes about 1.5% F_1, where the position-aware term in Eq. (3) alone contributes about 1% F_1 score.

Impact of negative examples. Figure 4 shows how the slot filling evaluation scores change as we change the amount of negative (i.e., *no_relation*) training data provided to our proposed model. We find that: (1) At hop-0 level, precision increases as we provide more negative examples, while recall stays almost unchanged. F_1 score keeps increasing. (2) At hop-all level, F_1 score increases by about 10% as we change the amount of negative examples from 20% to 100%.

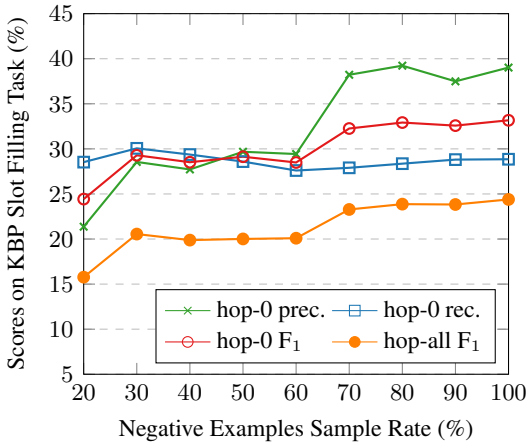

Figure 4: Change of slot filling hop-0 and hop-all scores as number of negative training examples changes. 100% is with all the negative examples included in the training set; the left side scores have positives and negatives roughly balanced.

Performance by sentence length. Figure 5 shows performance on varying sentence lengths. We find that: (1) Performance of all models degrades substantially as the sentences get longer. (2) Compared to the baseline Logistic Regression model, all neural models handle long sentences better. (3) Compared to CNN-PE model, RNN-based models are more robust on long sentences, and notably SDP-LSTM model is least sensitive to sentence length. (4) Our proposed model achieves equal or better results on sentences of all lengths, except for sentences with more than 60 tokens where SDP-LSTM model achieves the best result.

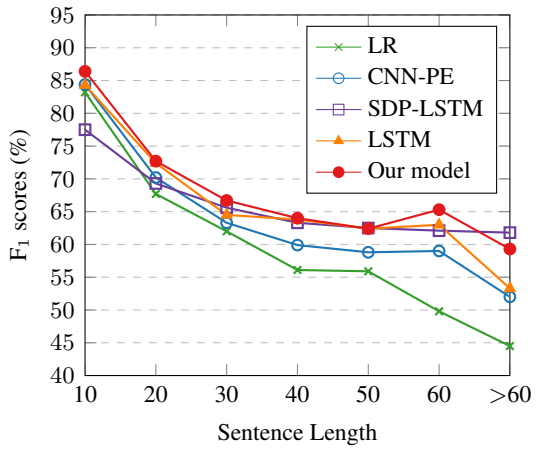

Figure 5: TACRED development set F_1 scores for sentences of varying lengths.

Improvement by slot types. We calculate the F_1 score for each slot type and compare the improvement from using our proposed model across slot types. When compared with the CNN-PE model, our position-aware attention model achieves improved F_1 scores on 30 out of the 41 slot types, with the top 5 slot types being *org:members*, *per:country_of_death*, *org:shareholders*, *per:children* and *per:religion*. When compared with SDP-LSTM model, our model achieves improved F_1 scores on 26 out of the 41 slot types, with the top 5 slot types being *org:political/religious_affiliation*, *per:country_of_death*, *org:alternate_names*, *per:religion* and *per:alternate_names*. We observe that slot types with relatively sparse training examples tend to be improved by using the position-aware attention model.

Attention visualization. Lastly, Figure 6 shows the visualization of attention weights assigned by our model on sampled sentences from the development set. We find that the model learns to pay more attention to words that are informative for the relation (e.g., "graduated from", "niece" and "chairman"), though it still makes mistakes (e.g., "refused to name the three"). We also observe that the model tends to put a lot of weight onto object entities, as the object NER signatures are very informative to the classification of relations.

5 Related Work

Relation extraction. There are broadly three main lines of work on relation extraction: first, fully-supervised approaches (Zelenko et al., 2003; Bunescu and Mooney, 2005), where a statistical classifier is trained on an annotated dataset; second, distant supervision (Mintz et al., 2009; Surdeanu et al., 2012), where a training set is formed by projecting the relations in an existing knowledge base onto textual instances that contain the entities that the relation connects; and third, Open IE (Fader et al., 2011; Mausam et al., 2012), which views its goal as producing subject-relation-object triples and expressing the relation in text.

Slot filling and knowledge base population. The most widely-known effort to evaluate slot filling and KBP systems is the yearly TAC KBP slot filling tasks, starting from 2009 (McNamee and Dang, 2009). Participants in slot filling tasks usually make use of hybrid systems that combine patterns, Open IE, distant supervision and supervised systems for relation extraction (Kisiel et al., 2015; Finin et al., 2015; Zhang et al., 2016).

Datasets for relation extraction. Popular general-domain datasets include the ACE dataset (Strassel et al., 2008) and the SemEval-2010 task 8 dataset (Hendrickx et al., 2009). In addition, the BioNLP Shared Tasks (Kim et al., 2009) are yearly efforts on creating datasets and evaluations for biomedical information extraction systems.

Deep learning models for relation extraction. Many deep learning models have been proposed for relation extraction, with a focus on end-to-end training using CNNs (Zeng et al., 2014; Nguyen and Grishman, 2015) and RNNs (Zhang et al., 2015). Other popular approaches include using CNN or RNN over dependency paths between entities (Xu et al., 2015a,b), augmenting RNNs with different components (Xu et al., 2016; Zhou et al., 2016), and combining RNNs and CNNs (Vu et al., 2016; Wang et al., 2016). Adel et al. (2016) compares the performance of CNN models against traditional approaches on slot filling using a portion of the TAC KBP evaluation data.

6 Conclusion

We introduce a state-of-the-art position-aware neural sequence model for relation extraction, as well as TACRED, a large-scale, crowd-sourced dataset that is orders of magnitude larger than previous relation extraction datasets. Our proposed model outperforms a strong feature-based classifier and all baseline neural models. In combination with the new dataset, it improves the state-of-the-

Sampled Sentences	Predicted Labels
PER-SUBJ graduated from North Korea 's elite Kim Il Sung University and ORG-OBJ ORG-OBJ .	*per:schools_attended*
The cause was a heart attack following a case of pneumonia , said PER-SUBJ 's niece , PER-OBJ PER-OBJ .	*per:other_family*
Independent ORG-SUBJ ORG-SUBJ ORG-SUBJ (ECC) chairman PER-OBJ PER-OBJ refused to name the three , saying they would be identified when the final list of candidates for the august 20 polls is published on Friday .	*org:top_members/employees*

Figure 6: Sampled sentences from the TACRED development set, with words highlighted according to the attention weights produced by our best model.

art hop-all F_1 on the TAC KBP 2015 slot filling task by 4.5% absolute.

Acknowledgments

We thank the anonymous reviewers for their helpful suggestions. We gratefully acknowledge the support of the Allen Institute for Artificial Intelligence and the support of the Defense Advanced Research Projects Agency (DARPA) Deep Exploration and Filtering of Text (DEFT) Program under Air Force Research Laboratory (AFRL) contract No. FA8750-13-2-0040. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, AFRL, or the US government.

References

Heike Adel, Benjamin Roth, and Hinrich Schütze. 2016. Comparing convolutional neural networks to traditional models for slot filling. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*.

Gabor Angeli, Sonal Gupta, Melvin Johnson Premkumar, Christopher D. Manning, Christopher Ré, Julie Tibshirani, Jean Y. Wu, Sen Wu, and Ce Zhang. 2014a. Stanford's distantly supervised slot filling systems for KBP 2014. In *Text Analysis Conference (TAC) Proceedings 2014*.

Gabor Angeli, Julie Tibshirani, Jean Y. Wu, and Christopher D. Manning. 2014b. Combining distant and partial supervision for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*.

Gabor Angeli, Victor Zhong, Danqi Chen, Arun Chaganty, Jason Bolton, Melvin Johnson Premkumar, Panupong Pasupat, Sonal Gupta, and Christopher D Manning. 2015. Bootstrapped self training for knowledge base population. In *Text Analysis Conference (TAC) Proceedings 2015*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*.

Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (EMNLP 2005)*, pages 724–731.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *Proceedings of the 53th Annual Meeting of the Association for Computational Linguistics (ACL 2015)*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Joe Ellis, Jeremy Getman, Dana Fore, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie Strassel. 2015. Overview of linguistic resources for the TAC KBP 2015 evaluations: Methodologies and results. In *Text Analysis Conference (TAC) Proceedings 2015*, pages 16–17.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pages 1535–1545.

Tim Finin, Dawn Lawrie, Paul McNamee, James Mayfield, Douglas Oard, Nanyun Peng, Ning Gao, Yiu-Chang Lin, Joshi MacKin, and Tim Dowd. 2015.

HLTCOE participation in TAC KBP 2015: Cold start and TEDL. In *Text Analysis Conference (TAC) Proceedings 2015*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of BioNLP'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 1–9.

Bryan Kisiel, Bill McDowell, Matt Gardner, Ndapandula Nakashole, Emmanouil A Platanios, Abulhair Saparov, Shashank Srivastava, Derry Wijaya, and Tom Mitchell. 2015. CMUML System for KBP 2015 cold start slot filling. In *Text Analysis Conference (TAC) Proceedings 2015*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534.

Paul McNamee and Hoa Trang Dang. 2009. Overview of the TAC 2009 knowledge base population track. In *Text Analysis Conference (TAC) Proceedings 2009*, volume 17, pages 111–113.

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

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*, pages 39–48.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, volume 14, pages 1532–1543.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL 2011)*, pages 1375–1384.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. *Machine learning and knowledge discovery in databases*, pages 148–163.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Stephanie Strassel, Mark A Przybocki, Kay Peterson, Zhiyi Song, and Kazuaki Maeda. 2008. Linguistic resources and evaluation techniques for evaluation of cross-document automatic content extraction. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016. Combining recurrent and convolutional neural networks for relation classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT)*.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015a. Semantic relation classification via convolutional neural networks with simple negative sampling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*.

Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. 2016. Improved relation classification by deep recurrent neural networks with

data augmentation. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015b. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 1785–1794.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research*, 3:1083–1106.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2014)*, pages 2335–2344.

Dongxu Zhang, Dong Wang, and Rong Liu. 2015. Relation classification via recurrent neural network. Technical report, CSLT 20150024, Tsinghua University.

Yuhao Zhang, Arun Chaganty, Ashwin Paranjape, Danqi Chen, Jason Bolton, Peng Qi, and Christopher D. Manning. 2016. Stanford at TAC KBP 2016: Sealing pipeline leaks and understanding chinese. In *Text Analysis Conference (TAC) Proceedings 2016*.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, page 207.

A TACRED Data Collection and Validation

In this appendix, we describe the way we collect and validate TACRED in full detail.

A.1 Data Collection

TACRED leverages the work done selecting query entities and annotating system responses in the TAC KBP evaluations. In each year of the TAC KBP evaluation (2009–2015), 100 query entities are given to participating KBP systems with the aim of filling in valid knowledge base entries for these entities. Our annotation effort re-uses these query entities, annotating each sentence in the source corpus that contains one of these entities. Given the set of mention pairs (e.g., *Penner* and *Lisa Dillman*) containing an evaluation entity, the mention pair can have either 1) been extracted during a previous KBP competition and marked correct by an LDC annotator, or 2) been generated automatically from candidate mention pairs in the corpus. For clarity, we refer to the former as **LDC examples** and the latter as **generated examples**, and describe them separately.

LDC examples. For examples in this category, although the relations have been annotated by an LDC annotator, the provenance for the mention pairs provided in TAC KBP evaluation files are often too general or imprecise; for example in early years only the document that contains a mention pair is given as provenance. We solve this problem with a two-stage annotation task (HIT) in Mechanical Turk: In the first task, Turk annotators are provided with the mention pair and its relation (annotated by LDC), and asked to find a sentence in the document that expresses the extraction. In the second task, annotators are asked to identify the spans of both the subject and object entities. See Figure 7 and Figure 8 for example interfaces provided to Turk annotators.

Generated examples. To further collect examples that are not annotated by LDC, we first run annotations on the corpus using a combination of Stanford's statistical coreference system (Clark and Manning, 2015) and the Illinois Wikifier (Ratinov et al., 2011). Then we collect all mention pairs in which one mention is linked to one of the query entities by the entity linker. To prevent the resulting dataset from being skewed towards commonly occurring query entities such as "Barack Obama", we enforce a hard upper limit on the number of collected mention pairs containing a query entity. Specifically, for each query entity q, we retrieve N_q sentences from the KBP corpus that contain an entity mention linked to q. Then let N_{q^c} denote the number of extractions submitted by competing KBP systems that were also deemed correct by human annotators, we want N_q to be proportional to N_{q^c}, and heuristically set: $N_q = \min(9 \cdot N_{q^c}, 300)$. Next, each mention pair, along with the corresponding sentence in which it occurs, is annotated for its relation type (or *no_relation*) as a task on Mechanical Turk. Figure 9 shows an example task interface for generated examples on Mechanical Turk.

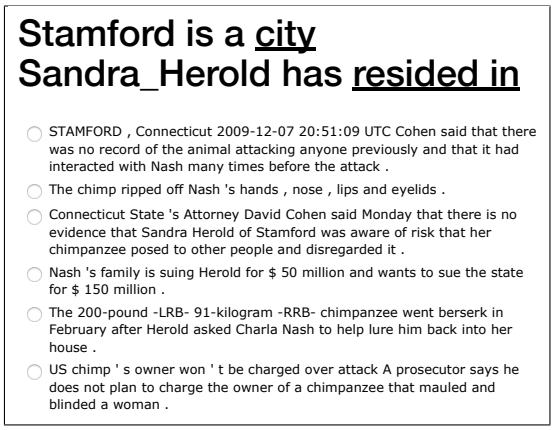

Figure 7: Example of an LDC examples HIT on Mechanical Turk for identifying the relevant sentence. The annotator is presented with every sentence from the document as well as the extraction for which to find the sentence.

A.2 Data Validation

In order to maintain the quality of TACRED, we validate the collected data both during and after the annotation process. We made use of crowdsourced data from a previous annotation effort on the same relation set (Angeli et al., 2014a). During annotation, 10% of the HITs presented to a worker are sanity check examples from this previous data, and annotators whose error rate on these examples exceeds 25% were asked to have their work re-annotated.

After the data collection is done, one of the authors manually examined 300 sampled instances. The estimated annotation accuracy is 93.3%, with a confidence interval of (89.9%, 95.9%). In addition, for the collected generated examples, we estimate inter-annotator agreement using 761 sampled

Figure 8: Example of an LDC examples HIT on Mechanical Turk for identifying the mention spans. The annotator is presented with a sentence obtained from the HIT shown in Figure 7 as well as the corresponding extraction and asked to identify the spans of the subject and object mentions in the extraction.

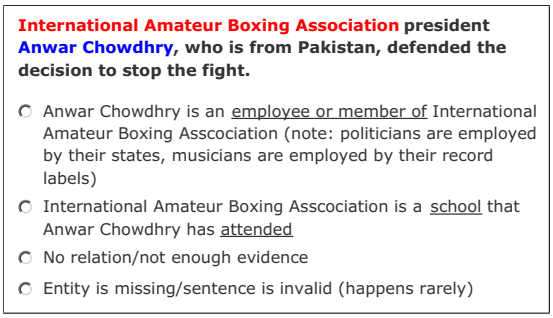

Figure 9: Example of a generated examples HIT. The subject entity is highlighted in blue and the object entity is highlighted in red. The annotator is asked to select among a set of plausible relations that are compatible with the subject and object entity types, along with an option to state that none of the presented relations hold.

mention pairs shown to five annotators. Results are shown in Table 7.

A.3 Data Statistics

In total, we collect 10,691 annotations from the LDC examples task and 110,021 annotations from the generated examples task. After removing examples where the subject and object entities overlap, we arrive at a total of 119,474 examples. About 78.7% of all examples are annotated as *no_relation*, which we showed to be crucial for training high-precision relation extraction models for the TAC KBP 2015 slot filling evaluation. Furthermore, we find that sentences in TACRED tend to be much longer than in the SemEval dataset

Metric	Score
5 annotators agree	74.2%
$\geq$ 4 annotators agree	90.5%
$\geq$ 3 annotators agree	100.0%
Fleiss Kappa	54.4%

Table 7: Estimated inter-annotator agreement using 761 sampled mention pairs.

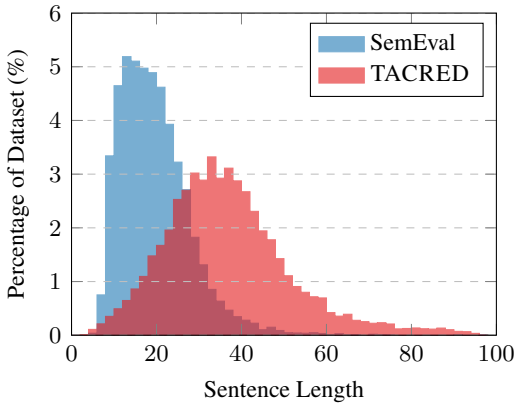

Figure 10: Distribution of sentence lengths in SemEval 2010 task 8 and TACRED.

(Figure 10).

Table 8 presents detailed statistics on this dataset. We also include sampled training examples in Table 9.

B Model Training Details

Here we describe the way we train our models in detail for replicability.

Model hyperparameters. We use 200 for word embedding size and 30 for every other embedding (i.e., position, POS or NER) size. For CNN models, we use filter window sizes ranging from 2 to 5, and 500 filters for each window size. For the SDP-LSTM model, in addition to POS and NER embeddings, we also include the type of dependency edges as an additional embedding channel. For our proposed position-aware neural sequence model, we use attention size of 200. For all models that require LSTM layers, we find a 2-layer stacked LSTMs works better than a single-layer LSTM. We use one-directional LSTM layers in all of our experiments. Empirically we find bi-directional LSTM layers give no improvement to our proposed position-aware sequence model and marginal improvement to the simple LSTM model. We do not add max-pooling layers after

LSTM layers as we find this harms the performance.

Training. During training, we employ standard dropout (Srivastava et al., 2014) for CNN models, and RNN dropout (Zaremba et al., 2014) for LSTM models. Additionally, for CNN models we apply ℓ_2 regularization with coefficient 10^{-3} to all filters to avoid overfitting. We use *AdaGrad* (Duchi et al., 2011) with a learning rate of 0.1 for CNN models and 1.0 for all other models. We train CNN models for 50 epochs and other models for 30 epochs, with a mini-batch size of 50. We monitor the training process by looking at the micro-averaged F_1 score on the dev set. Starting from the 20th epoch, we decrease the learning rate with a decay rate of 0.9 if the dev set micro-averaged F_1 score does not increase after every epoch. Finally, we evaluate the model that achieves the best dev set F_1 score on the test set.

Relation	Total	Percentage	Train 2009–2012	Development 2013	Test 2014
no_relation	94001	78.68%	60179	19305	14517
org:alternate_names	1515	1.27%	893	380	242
org:city_of_headquarters	656	0.55%	437	125	94
org:country_of_headquarters	878	0.73%	540	215	123
org:dissolved	41	0.03%	29	8	4
org:founded	199	0.17%	103	49	47
org:founded_by	343	0.29%	145	109	89
org:member_of	222	0.19%	147	39	36
org:members	330	0.28%	194	95	41
org:number_of_employees/members	144	0.12%	87	35	22
org:parents	528	0.44%	332	120	76
org:political/religious_affiliation	148	0.12%	118	13	17
org:shareholders	168	0.14%	87	66	15
org:stateorprovince_of_headquarters	407	0.34%	266	83	58
org:subsidiaries	516	0.43%	326	138	52
org:top_members/employees	3182	2.66%	2138	635	409
org:website	302	0.25%	133	133	36
per:age	977	0.82%	416	292	269
per:alternate_names	172	0.14%	111	48	13
per:cause_of_death	384	0.32%	127	199	58
per:charges	322	0.27%	77	120	125
per:children	385	0.32%	235	109	41
per:cities_of_residence	857	0.72%	421	203	233
per:city_of_birth	126	0.11%	77	40	9
per:city_of_death	271	0.23%	102	133	36
per:countries_of_residence	978	0.82%	498	281	199
per:country_of_birth	74	0.06%	39	26	9
per:country_of_death	83	0.07%	10	57	16
per:date_of_birth	127	0.11%	78	39	10
per:date_of_death	451	0.38%	151	238	62
per:employee_of	2621	2.19%	1837	433	351
per:origin	794	0.66%	373	257	164
per:other_family	417	0.35%	233	96	88
per:parents	334	0.28%	164	59	111
per:religion	186	0.16%	61	65	60
per:schools_attended	277	0.23%	178	62	37
per:siblings	284	0.24%	178	37	69
per:spouse	569	0.48%	311	185	73
per:stateorprovince_of_birth	88	0.07%	47	30	11
per:stateorprovince_of_death	133	0.11%	65	53	15
per:stateorprovinces_of_residence	560	0.47%	374	89	97
per:title	4424	3.70%	2733	1065	626
Total	119474	100.00%	75050	25764	18660

Table 8: Relation distribution of the TACRED dataset.

Example Sentences	Subject Type	Object Type	Relation Labels
Carey will succeed **Cathleen P. Black**, who held the position for 15 years and will take on a new role as *chairwoman* of Hearst Magazines, the company said.	Person	Title	per:title
Baldwin declined further comment, and said JetBlue chief *executive* Dave Barger was unavailable.	Person	Title	no_relation
Irene Morgan Kirkaldy, who was born and reared in Baltimore, lived on Long Island and ran a child-care center in Queens with her second husband, *Stanley Kirkaldy*.	Person	Person	per:spouse
Cummings, current holder of the Seventh District seat held by *Mr. Mitchell*, sponsored legislation last year that named a Baltimore post office in the veteran congressman's honor.	Person	Person	no_relation
Blackburn Rovers announced Tuesday they had sacked *Paul Ince* as their manager, a statement on the Premier League club's website said.	Organization	Person	org:top_members/employees
Kerry wrote a letter to *Pickens*, saying he would donate any proceeds to the **Paralyzed Veterans of America**, the Associated Press reported.	Organization	Person	no_relation
Forsberg launched the anti-nuclear movement with a paper she wrote while obtaining a doctorate in international studies at *Massachusetts Institute of Technology*.	Person	Organization	per:schools_attended
He received an undergraduate degree from Morgan State University in 1950 and applied for admission to graduate school at the *University of Maryland in College Park*.	Person	Organization	no_relation

Table 9: Sampled training examples from the TACRED dataset, with subject entity highlighted in bold and blue and object entities highlighted in italics and red.