Combining Word Embeddings and Feature Embeddings for Fine-grained Relation Extraction

Mo Yu
Machine Intelligence & Translation Lab
Harbin Institute of Technology
Harbin, China
gflfof@gmail.com

Matthew R. Gormley, Mark Dredze
Human Language Technology Center of Excellence
Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD, 21218
{mgormley, mdredze}@cs.jhu.edu

Abstract

Compositional embedding models build a representation for a linguistic structure based on its component word embeddings. While recent work has combined these word embeddings with hand crafted features for improved performance, it was restricted to a small number of features due to model complexity, thus limiting its applicability. We propose a new model that conjoins features and word embeddings while maintaining a small number of parameters by learning feature embeddings jointly with the parameters of a compositional model. The result is a method that can scale to more features and more labels, while avoiding overfitting. We demonstrate that our model attains state-of-the-art results on ACE and ERE fine-grained relation extraction.

1 Introduction

Word embeddings represent words in some low-dimensional space, where each dimension might intuitively correspond to some syntactic or semantic property of the word. These embeddings can be used to create novel features (Miller et al., 2004; Koo et al., 2008; Turian et al., 2010; Sun et al., 2011; Nguyen and Grishman, 2014; Roth and Woodsend, 2014), and can also be treated as model parameters to build representations for higher-level structures in some compositional embedding models (Collobert et al., 2011; Collobert, 2011; Socher et al., 2012; Socher et al., 2013b; Hermann et al., 2014). Applications of embedding have boosted the performance of many NLP tasks, including syntax (Turian et al., 2010; Collobert et al., 2011), semantics (Socher et al., 2012; Socher et al., 2013b; Hermann et al., 2014), question answering (Bordes et al., 2014) and machine translation (Devlin et al., 2014).

While compositional models aim to learn higher-level structure representations, composition of embeddings alone may not capture important syntactic or semantic patterns. Consider the task of relation extraction, where decisions require examining long-distance dependencies in a sentence. For the sentence in Figure 1, “driving” is a strong indicator of the “ART” (ACE) relation because it appears on the dependency path between a person and a vehicle. Yet such conjunctions of different syntactic/semantic annotations (dependency and NER) are typically not available in compositional models.

In contrast, hand-crafted features can easily capture this information, e.g. feature \( f_{13} \) (Figure 1). Therefore, engineered features should be combined with learned representations in compositional models. One approach is to use the features to select specific transformations for a sub-structure (Socher et al., 2013a; Hermann and Blunsom, 2013; Hermann et al., 2014; Roth and Woodsend, 2014), which can conjoin features and word embeddings, but is impractical as the numbers of transformations will exponentially increase with additional features. Typically, less than 10 features are used. A solution
Table 7: Performance on ACE2005 test sets. The first part of the table shows the performance of different models on

| Model                  | Dev MRR | Test MRR |
|------------------------|---------|----------|
| FCT                    | 41.96   | 39.10    |
| FCN                    | 42.78   | 39.25    |
| FCN                          | 43.1    | 39.9     |
| Recursive NN                | 45.0    | 41.0     |
| SUM                    | 46.95   | 43.4     |

Figure 1: Example of input structure. Left: a sentence with target entities \( \{M_1, M_2\} \) and annotations \( A \) (e.g. dependency tree). Right: outer product representation of a single word \( w_i \) with an example of useful features \( f_i \).

We present a new method of learning interactions between engineered features and word embeddings by combining the idea of the outer product in FCM (Yu et al., 2014) with learning feature embeddings (Collobert et al., 2011; Chen and Manning, 2014). Our model jointly learns feature embeddings and a tensor-based classifier which relies on the outer product between features embeddings and word embeddings. Therefore, the number of parameters are dramatically reduced since features are only represented as low-dimensional embeddings, which alleviates problems with overfitting. The resulting model benefits from both approaches: conjunctions between feature and word embeddings allow model expressiveness, while keeping the number of parameters small. This is especially beneficial when considering tasks with many labels, such as fine-grained relation extraction. We demonstrate these advantages on two relation extraction tasks: the well-studied ACE 2005 dataset and the new ERE relation extraction task. We consider both coarse and fine-grained relations, the latter of which has been largely unexplored in previous work.

2 Factor-based Compositional Embedding Models (FCM)

We begin by briefly summarizing the FCM model proposed by Yu et al. (2014) in the context of relation extraction. In relation extraction, for a pair of mentions in a given sentence, the task is to determine the type of relation that holds between the two entities, if any. For each pair of mentions in a sentence, we have a training instance \((x, y)\); \(x\) is an annotated sentence, including target entity mentions \(M_1\) and \(M_2\), and a dependency parse. We consider directed relations: for relation type \(Rel, y = Rel(M_1, M_2)\) and \(y' = Rel(M_2, M_1)\) are different.

FCM has a log-linear form, which defines a particular utilization of the features and embeddings. FCM decomposes the structure of \(x\) into single words. For each word \(w_i\), a binary feature vector \(f_i\) is defined, which considers the \(i\)th word and any other substructure of the annotated sentence \(x\). We denote the dense word embedding by \(e_w\) and the label-specific model parameters by matrix \(T_y\), e.g. in Figure 1, the gold label corresponds to matrix \(T_y\) where \(y = ART(M_1, M_2)\). FCM is then given by:

\[
P(y|x; T) \propto \exp(\sum_i T_y \odot (f_i \otimes e_w)) \tag{1}
\]

where \(\odot\) is the outer-product of the two vectors and \(\otimes\) is the ‘matrix dot product’ or Frobenious inner product of the two matrices. Here the model parameters form a tensor \(T = [T_1 : \ldots : T_{|L|}]\), which transforms the input matrix to the labels.

The key idea in FCM is that it gives similar words (i.e. those with similar embeddings) with similar functions in the sentence (i.e. those with similar features) similar matrix representations. Thus, this model generalizes its model parameters across words with similar embeddings only when they share similar functions in the sentence. For the
example in Figure 1, FCM can learn parameters which give words similar to “driving” with the feature $f_3 = 1$ (is-on-dependency-path $\land$ type($M_1$) = PER $\land$ type($M_2$) = VEH) high weight for the ART label.

3 Low-Rank Approximation of FCM

FCM achieved state of the art performance on SemEval relation extraction (Yu et al., 2014), yet its generalization ability is limited by the size of the tensor $T$, which cannot easily scale to large number of features. We propose to replace features with feature embeddings (Chen and Manning, 2014), thereby reducing the dimensionality of the feature space, allowing for more generalization in learning the tensor. This will be especially beneficial with an increased number of output labels (i.e. more relation types), as this increases the number of parameters.

Our task is to determine the label $y$ (relation) given the instance $x$. For each word $w_i \in x$, there exists a list of $m$ associated features $f_i = f_{i,1}, f_{i,2}, ..., f_{i,m}$. The model then transforms the feature vector into a $d_g \times m$ vector with a matrix (i.e. a lookup table) $W_f$ as: $g_i = f_i \cdot W_f$. Here we use a linear transformation for computational efficiency. We score label $y$ given $x$ as (replacing Eq. 1):

$$P(y|x; T, W_f) \propto \exp(\sum_y T_{y} \odot (g_i \odot e_{w_i}))$$ (2)

We call this model low-rank FCM (LRFCM). The result is a dramatic reduction in the number of model parameters, from $O(md|L|)$ to $O(d_g d|L| + d_g m)$, where $d$ is the size of the word embeddings. This reduction is intended to reduce the variance of our estimator, possibly at the expense of higher bias. Consider the case of 32 labels (fine-grained relations in §4), 3,000 features, and 200 dimensional word embeddings. For FCM, the size of $T$ is $1.92 \times 10^7$; potentially yielding a high variance estimator. However, for LRFCM with 20-dimensional feature embeddings, the size of $T$ is $1.28 \times 10^5$, significantly smaller with lower variance. Moreover, feature embeddings can capture correlations among features, further increasing generalization.

Figure 2 shows the vectorized form of LRFCM as a multi-layer perceptron. LRFCM constructs a dense low-dimensional matrix used as the input to Eq. 2. By contrast, FCM does not have a feature embedding layer and both feature vector $f$ and word embedding $e_w$ are feed forward directly to the outer product layer.

**Training** We optimize the following log-likelihood (of the probability in Eq. 2) objective with AdaGrad (Duchi et al., 2011) and compute gradients via back-propagation:

$$\mathcal{L}(T, W_f) = \frac{1}{|D|} \sum_{(y, x) \in D} \log P(y|x; T, W_f),$$ (3)

where $D$ is the training set. For each instance $(y, x)$ we compute the gradient of the log-likelihood $\ell = \log P(y|x; T, W_f)$. We define the vector $s = [(\sum_i T_{y i} \odot (g_i \odot e_{w_i}))|1 \leq y \leq L]^T$, which yields $\partial \ell / \partial s = [(I[y = y'] - P(y'|x; T, W_f))|1 \leq y' \leq L]^T$, where $I[x]$ is the indicator function equal to 1 if $x$ is true and 0 otherwise. Then we have the following stochastic gradients, where $\circ$ is the tensor product:

$$\frac{\partial \ell}{\partial T} = \frac{\partial \ell}{\partial s} \odot \sum_{i=1}^{n} g_i \odot e_{w_i},$$ (4)

$$\frac{\partial \ell}{\partial W_f} = \sum_{i=1}^{n} \frac{\partial \ell}{\partial g_i} \odot g_i \odot W_f = \sum_{i=1}^{n} \left(T \odot \frac{\partial \ell}{\partial s} \odot e_{w_i}\right) \odot f_i.$$ (5)

4 Experiments

**Datasets** We consider two relation extraction datasets: ACE2005 and ERE, both of which contain two sets of relations: coarse relation types and fine relation (sub-)types. Prior work on English ACE 2005 has focused only on coarse relations (Plank and Moschitti, 2013; Nguyen and Grishman, 2014; Li and Ji, 2014); to the best of our knowledge, this paper establishes the first baselines for the other datasets. Since the fine-grained relations require a large number of parameters, they will test the ability
Table 1: Results on test for ACE and ERE where only the entity spans (S) are known (top) and where both the entity spans and types are known (ST). PM’13 is an embedding method. The sizes of relation sets are indicated by $|L|$. PM’13 (S) 55.3 43.1 48.5 - - - - - - - - - -

| Model       | ACE-bc ($|L|=11$) | ACE-bc ($|L|=32$) | ERE ($|L|=9$) | ERE ($|L|=18$) |
|-------------|------------------|------------------|--------------|--------------|
|             | P R F1           | P R F1           | P R F1       | P R F1       |
| PM’13 (S)   | 55.3 43.1 48.5   | - - -            | - - -        | - - -        |
| FCM (S)     | 62.3 45.1 52.3   | 59.7 41.6 49.0   | 68.3 52.5 59.4 | 67.1 51.5 58.2 |
| LRFCM(S)    | 58.5 46.8 52.0   | 57.4 46.2 51.2   | 65.1 56.1 60.3 | 65.4 55.3 59.9 |
| Baseline (ST) | 72.2 52.0 60.5   | 60.2 51.2 55.3   | 76.2 64.0 69.5 | 73.5 62.1 67.3 |
| FCM (ST)    | 66.2 54.2 59.6   | 62.9 49.6 55.4   | 73.0 65.4 69.0 | 74.0 60.1 66.3 |
| LRFCM (ST)  | 65.1 54.7 59.4   | 63.5 51.1 56.6   | 75.0 65.7 70.0 | 73.2 63.2 67.8 |

of LRFCM to scale and generalize. As is standard, we report precision, recall, and F1 for all tasks.

**ACE 2005** We use the English portion of the ACE 2005 corpus (Walker et al., 2006). Following Plank and Moschitti (2013), we train on the union of the news domains (Newswire and Broadcast News), hold out half of the Broadcast Conversation (bc) domain as development data, and evaluate on the remainder of bc. There are 11 coarse types and 32 fine (sub-)type classes in total. In order to compare with traditional feature-based methods (Sun et al., 2011), we report results in which the gold entity spans and types are available at both train and test time. We train the models with all pairs of entity mentions in the training set to yield 43,518 classification instances. Furthermore, for comparison with prior work on embeddings for relation extraction (Plank and Moschitti, 2013), we report results using gold entity spans but no types, and generate negative relation instances from all pairs of entities within each sentence with three or fewer intervening entities.

**ERE** We use the third release of the ERE annotations from Phase 1 of DEFT (LDC, 2013). We divided the proxy reports summarizing news articles (pr) into training (56,889 relations), development (6,804 relations) and test data (6,911 relations). We run experiments under both the settings with and without gold entity types, while generating negative relation instances just as in ACE with the gold entity types setting. To the best of our knowledge, we are the first to report results on this task.

Following the annotation guidelines of ERE relations, we treat all relations, except for “social.business”, “social.family” and “social.unspecified”, as asymmetric relations. For coarse relation task, we treat all relations as asymmetric, including the “social” relation. The reason is that the asymmetric subtype, “social.role”, dominates the class: 679 of 834 total “social” relations.

**Setup** We randomly initialize the feature embeddings $W_f$ and pre-train 200-dimensional word embeddings on the NYT portion of Gigaword 5.0 (Parker et al., 2011) with word2vec (default setting of the toolkit) (Mikolov et al., 2013). Dependency parses are obtained from the Stanford Parser (De Marneffe et al., 2006). We use the same feature templates as Yu et al. (2014). When gold entity types are unavailable, we replace them with WordNet tags annotated by Ciaramita and Altun (2006). Learning rates, weights of L2-regularizations, the number of iterations and the size of the feature embeddings $d$ are tuned on dev sets. We selected $d$ from $\{12, 15, 20, 25, 30, 40\}$. We used $d=30$ for feature embeddings for fine-grained ACE without gold types, and $d=20$ otherwise. For ERE, we have $d=15$. The weights of L2 $\lambda$ was selected from $\{1e-3, 5e-4, 1e-4\}$. As in prior work (Yu et al., 2014), regularization did not significantly help FCM. However for LRFCM, $\lambda=1e-4$ slightly helps. We use a learning rate of 0.05.

We compare to two baselines. First, we use the features of Sun et al. (2011), who build on Zhou et al. (2005) with additional highly tuned features for ACE-style relation extraction from years of research. We implement these in a logistic regression model BASELINE, excluding country gazetteer and WordNet features. This baseline includes gold entity types and represents a high quality feature rich model. Second, we include results from Plank and Moschitti (2013) (PM’13), who obtained improve-
Table 2: Confusion Matrix between the results of FCM and LRFCM on the test set of ERE fine relation task. Each item in the table shows the number of relations on which the two models make correct/incorrect predictions.

|       | FCM Correct | Incorrect |
|-------|-------------|-----------|
| FCM   | 423         | 34        |
| Incorrect | 57    | 246       |

Table 2: Confusion Matrix between the results of FCM and LRFCM on the test set of ERE fine relation task. Each item in the table shows the number of relations on which the two models make correct/incorrect predictions.

Results

Both FCM and LRFCM outperform Plank and Moschitti (2013) (no gold entities setting) (Table 1). With gold entity types, the feature-rich baseline beats both composition models for ACE coarse types. However, as we consider more labels, LRFCM improves over this baseline, as well as for ERE coarse types. Furthermore, LRFCM outperforms FCM on all tasks, save ACE coarse types, both with and without gold entity types. The fine-grained settings demonstrate that our model can better generalize by using relatively fewer parameters. Additionally, the gap between train and test F1 makes this clear. For coarse relations, FCM’s train to test F1 gap was 35.2, compared to LRFCM with 25.4. On fine relations, the number increases to 40.2 for FCM but only 31.2 for LRFCM. In both cases, LRFCM does not display the same degree of overfitting.

Analysis

To highlight differences in the results we provide the confusion matrix of the two models on ERE fine relations. Table 2 shows that the two models are complementary to each other to a certain degree. It indicates that the combination of FCM and LRFCM may further boost the performance. We leave the combination of FCM and LRFCM, as well as their combination with the baseline method, to future work.

5 Conclusion

Our LRFCM learns conjunctions between features and word embeddings and scales to many features and labels, achieving improved results for relation extraction tasks on both ACE 2005 and ERE.

To the best of our knowledge, we are the first to report relation extraction results on ERE. To make it easier to compare to our results on these tasks, we make the data splits used in this paper and our implementation available for general use.

Acknowledgements

Mo Yu is supported by China Scholarship Council and by NSFC 61173073.

References

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Frédéric Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In Innovations in Machine Learning, pages 137–186. Springer.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. arXiv preprint arXiv:1406.3676.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 740–750, Doha, Qatar, October. Association for Computational Linguistics.

Massimiliano Ciaramita and Yasemin Altun. 2006. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In EMNLP2006, pages 594–602, July.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. JMLR, 12:2493–2537.

Ronan Collobert. 2011. Deep learning for efficient discriminative parsing. In International Conference on Artificial Intelligence and Statistics.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In Proceedings of LREC, volume 6, pages 449–454.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhou. 2014. Fast and robust neural network joint models for statistical machine translation. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1370–1380, Baltimore, Maryland, June. Association for Computational Linguistics.

We used their implementation: https://github.com/Gorov/FCM_nips_workshop/

https://github.com/Gorov/ERE_RE
