Mapping Unseen Words to Task-Trained Embedding Spaces

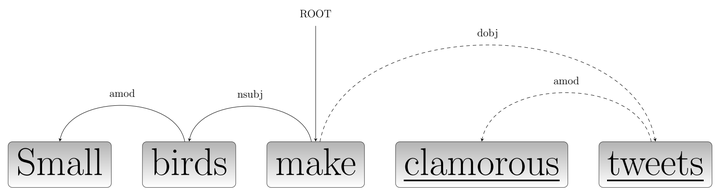
Pranava Madhyastha*, Mohit Bansal†♯, Kevin Gimpel†, Karen Livescu†

*Universitat Politècnica de Catalunya, Barcelona pranava@cs.upc.edu

†Toyota Technological Institute at Chicago {mbansal,kgimpel,klivescu}@ttic.edu

♯ Now at UNC Chapel Hill

Motivation



- Setting: Train a Supervised Neural Network Based Parser
- Initialize with pre-trained word embeddings
- Fine-tune the word embeddings while training the parser
- Problem: At test-time, unseen words are not in task-trained training embedding space
- Typical solution: use pre-trained embeddings or a single unknown word vector. Neither is ideal!

Key Idea

Mapping Initial Word Embeddings of Unseen Test-Time Words to Task-Trained Embedding Space

- Neural network mapper with a *weighted multiple-loss criterion*.
- Tune mapper's hyperparameters to optimize performance on each domain.
- Significant improvements in dependency parsing across several domains and downstream task of dependency-based sentiment analysis

Training Pipeline

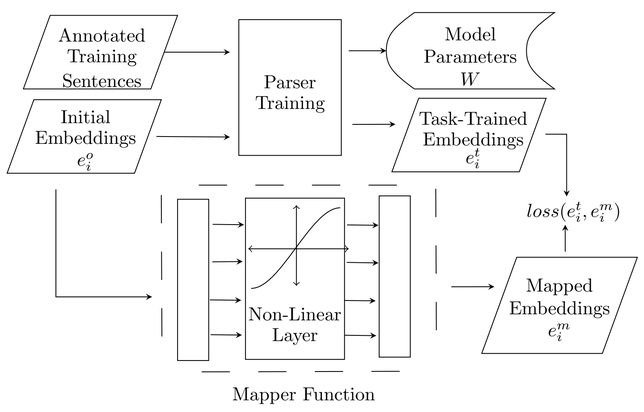


Figure: Mapper Training

— Basic Mapper Framework:

$$e_i^m = W_2(\text{hard}\tanh(W_1 e_i^o + b_1)) + b_2$$

— Weighted, multi-loss regression:

$$loss(y, \hat{y}) = \alpha \sum_{j=1}^{n} |y_j - \hat{y}_j| + (1-\alpha) \sum_{j=1}^{n} |y_j - \hat{y}_j|^2$$

— Elastic-net Regularization:

$$\lambda_1 \|\theta\|_1 + \frac{\lambda_2}{2} \|\theta\|_2^2$$

Parsing Pipeline

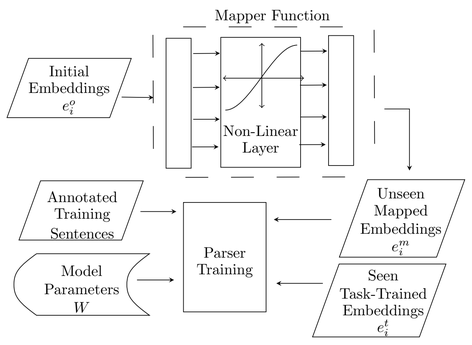


Figure: Test-time: Parsing with Mapped Embeddings

— Trained Mapper transforms original embeddings.

— Mapping Thresholds:

1. Mapper-training Threshold (τ_t): Only train the mapper on words that appear >= τ_t times.
2. Mapping Threshold (τ_m): Map parser-trained embeddings that appear <= τ_m times.
3. Parser Threshold (τ_p): Parser only uses embeddings of words that occur >= τ_p times.

Experimental Settings and Data

Dependency Parser: Standard feed-foward neural network based parser from Chen & Manning (2014)

Word Embeddings: 100−dimensional pre-trained GloVe word embeddings from Pennington et al. (2014).

Data: WSJ portion of Penn Treebank: Standard splits and predicted POS Tags.

Web Treebank: Train using OntoNotes-WSJ and testing on Web Treebank; We use a small split of the testset as held-out data for mapper.

Downstream Task: Experiments on Sentiment Analysis task, performed on Tai et al. (2015)'s Dependency Tree LSTM. Replace basic dependency parser with mapper enhanced version.

Internals

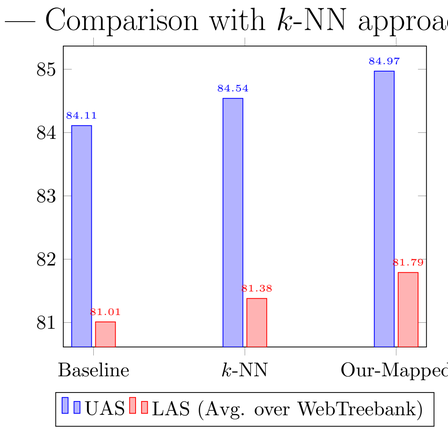
- Optimization using L-BFGS.
- α and λs are tuned using held-out domain specific data.
- Initial embeddings to both the parser and the mapper are the same.
- Dimensionality of the non-linear layer is set to 400.
- Hyperparameters give us additional flexibility to map the embeddings for the domain of interest, especially when the training dataset and testing dataset are from different domains.

Experiments

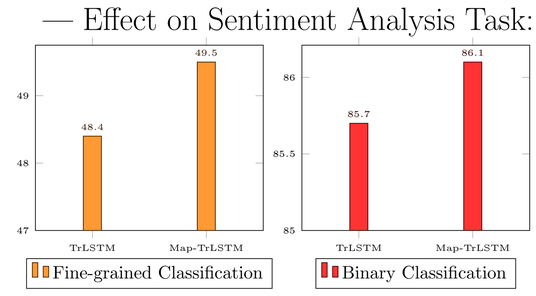
— Experiments on WSJ and Web Treebank

	Lower OOTV word rate		Higher OOTV word rate				
	WSJ	OntoNotes	Answers	Emails	Newsgroups	Reviews	Weblogs
UAS	91.85→92.21	90.17→90.49	82.67→83.21	81.76→82.42	84.68→85.13	84.25→84.99	87.73→88.43
OOTV %	2.72→1.45	2.72→1.4	8.53→1.22	10.56→3.01	10.34→1.04	6.84→0.73	8.45→0.38
OOTV UAS	89.88→90.51	89.27→89.81	80.88→81.75	79.29→81.02	82.54→83.71	81.17→82.22	86.43→87.31

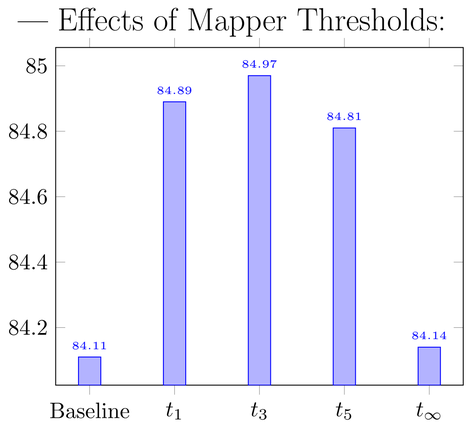
— Comparison with k-NN approach:



— Effect on Sentiment Analysis Task:



— Effects of Mapper Thresholds:



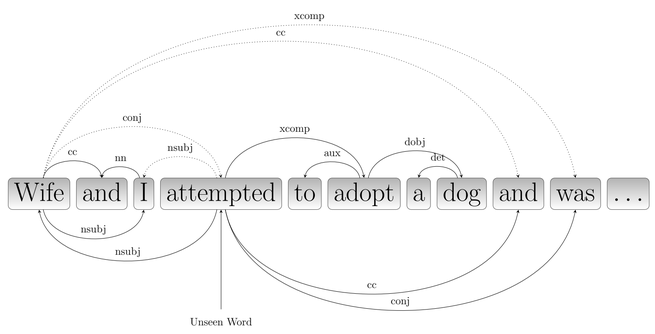
- We consider the threshold in the following settings:

$$\boldsymbol{t_1} : \tau_m = \tau_t = \tau_p = 1$$
$$\boldsymbol{t_3} : \tau_m = \tau_t = \tau_p = 3$$
$$\boldsymbol{t_5} : \tau_m = \tau_t = \tau_p = 5$$
$$\boldsymbol{t_\infty} : \tau_m = \infty, \tau_p = \tau_t = 5$$

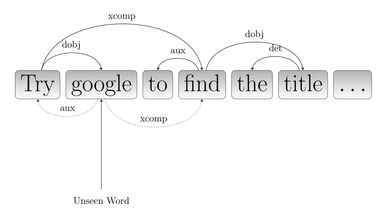
Using $\tau_m = \infty$ corresponds to mapping all words at test time.

What Works and What Doesn't

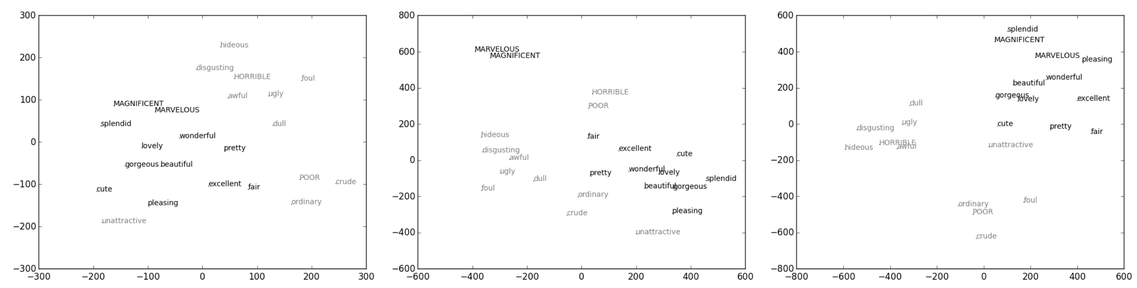
Mapper helps in obtaining the correct tree.



Mapper incorrectly maps 'google'



Representational Spaces



Initial Space | Learned Space | Mapped Space

Observations and Conclusions

- A simple method to resolve unseen words when training supervised models that learn task-specific word embeddings:
- Significant improvements in dependency parsing accuracy across several domains, as well as improvements on a downstream task
- Approach is effective, and applicable to many other settings, both inside and outside NLP.

Acknowledgements: Work when the first author was a visiting student at TTI-Chicago. This research was supported by a Google Faculty Research Award to Mohit Bansal, Karen Livescu and Kevin Gimpel.