1918 TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

INSTITUTE OF CYBERNETICS

Multi-domain Neural Network Language Model

Tanel Alumäe
Institute of Cybernetics at Tallinn University of Technology, Estonia

Summary

We describe a neural network language model that jointly models language in many related domains. The model uses the ID of the domain as an additional input to the neural network. Adaptation is implemented by training a vector of factors for each domain that is used to modulate the connections from the projection layer to the hidden layer. This eliminates the need to optimize a separate model for each target domain. Experiments show that this approach can improve LM perplexity and reduce speech recognition error rates.

Motivation

Language model is usually estimated from **several training sources (domains)**, e.g.:

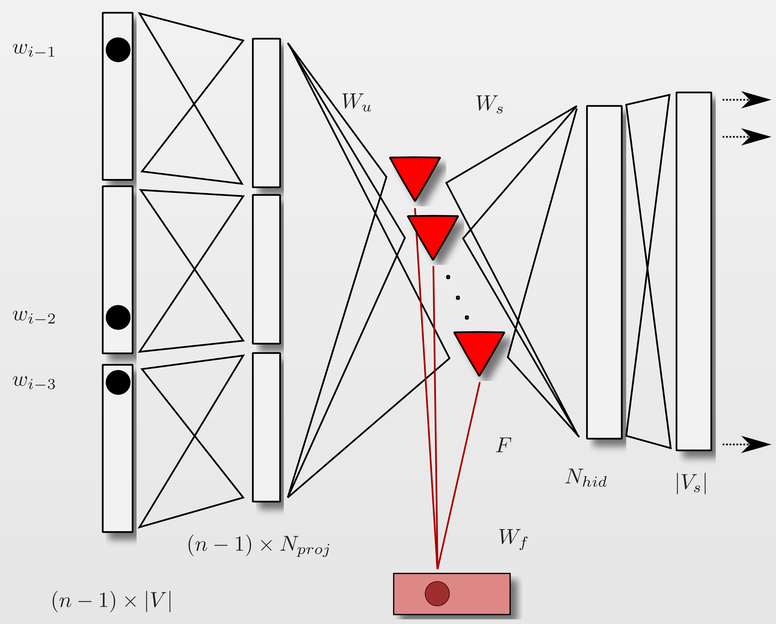
- newspaper texts
- web data
- human-transcribed speech

N-gram models are usually built by interpolating domain-specific models, using optimized coefficients. However, this approach is impractical for NNLMs.
Traditional approaches for combining multiple training domains for NNLMs:

1. Concatenate all training data: usually we have much more data from written domains (e.g., newspapers), and the resulting model will be biased towards written domains.
2. Sample data during training, based on data importance – at each epoch, use all data from the target domain, and a small random subset of other domains.
3. Adaptation: first, build a model as in (1), and then add an additional *adaptation layer*, trained using only in-domain data.

The proposed approach models multiple domains jointly. The domain of the text is simply used as an **additional input** to the neural network LM, that **scales the interactions** between the projection and the hidden layer. The model is jointly optimized for all domains. Domain is selected during training/testing using additional model input.

Architecture



d_i – current domain

Model description

Traditional neural network LM consists of:

1. binary input layer, representing the indexes of $n-1$ previous words
2. linear (tied) projection layer with $(n-1) \times |V|$ inputs and $(n-1) \times N_{proj}$ output
3. non-linear hidden layer of with N_{hid} units
4. linear output layer with softmax function.

Multi-domain model adds a new adaptation layer between (2) and (3). The adaptation layer consists of F **factors** (typically, $F = 300..500$). For each factor we do the following:

- compute a weighted sum of the projection layer outputs, using a factor-specific and domain-independent weight vector;
- the resulting sum is multiplied by a domain- and factor-specific scalar (+bias);
- the resulting scalars are fed to the hidden layer, using domain-independent weights.

...

The interaction between projection and hidden layer is thus determined by three matrices:

1. W_u (dimensions: $N_{proj} \times (n-1) \times F$) is shared across domains
2. W_f (dimensions: $(N_{dom}+1) \times F$) has a F-dimensional vector for each domain
3. W_s (dimensions: $F \times N_{hid}$) is shared across domains

Weighted input vector to the hidden layer z_{hidden} is calculated from the output vector of the projection layer y_{proj} as follows:

$$z_{hidden} = \left[\left(y_{proj}^T W_u \odot \left(w_f^{(d_i)} + w_f^{(bias)} \right) \right) W_s \right]^T$$

where $\odot$ is elementwise multiplication and d_i is the domain of the text.
Intuition: domain-specific vector $w_f^{(d_i)}$ amplify or attenuate connections between projection and hidden layer.

Experiments

Data:

- **Estonian**: 60k vocabulary. Newspapers (1.4M words), Web (1.7M), Broadcast News (BN) (133k), Broadcast Conversations (BC) (293k)
- **English**: 26k vocabulary. Newswire (13M), BN (950K)
- **French**: 40k vocabulary. Newspapers (4.5M), BN (500k), BC (300k)

Setup:

- All experiments used 4-gram models.
- Baseline N-gram: Kneser-Ney smoothed LM built by interpolating source-specific N-gram models, interpolation coefficients optimized on dev data.
- For comparison: adapted maximum entropy (MaxEnt) model, built by first estimating a background MaxEnt model over all data, and then adapting it to a particular domain.
- All NNLMs use a shortlist of 1k words, optimized N-gram used for out-of-shortlist words

Perplexity evaluations:

Model	*Estonian* BN	BC	*English* BN	*French* BN	BC	*Improvement over N-gram*
N-gram	351	434	252	143	141	
Adapted MaxEnt	310	405	234	133	**126**	-8.6%
Simple NNLM	332	454	253	137	147	-0.1%
Multi-domain NNLM	310	405	224	129	130	-9.4%
Simple NNLM + N-gram	310	406	224	129	130	-9.4%
Multi-domain NNLM + N-gram	**293**	**386**	**218**	**127**	**126**	**-12.6%**

Speech recognition experiments. Based on lattice rescoring using the particular models. Lattices were generated using a three-pass transcription system, with CMLLR and MLLR adaptation between the passes. Only Estonian speech data was available for experiments.

Model	*Estonian* BN	BC	*Relative improvement over N-gram*
N-gram	19.4	34.8	
Adapted MaxEnt	19.2	33.8	-2.0%
Simple NNLM + N-gram	18.7	34.1	-2.8%
Multi-domain NNLM + N-gram	**18.5**	**33.6**	**-4.0%**

Conclusion

- Multi-domain NNLM models jointly many text domains. Current domain is selected using additional model input.
- Domain-specific differences are integrated to the model using scaled connections between projection and hidden layer.
- Adds about 20% complexity during training.
- Can be easily combined with more advanced NNLM architectures (e.g., structured output layer, dropout).