Sparse Non-negative Matrix Language Modeling

Joris Pelemans
joris@pelemans.be

Noam Shazeer
noam@google.com

Ciprian Chelba
ciprianchelba@google.com

Outline

- Motivation
- Sparse Non-negative Matrix Language Model
- Skip-grams
- Experiments, investigating:
  - Modeling Power (sentence level)
  - Computational Complexity
  - Cross-sentence Modeling
  - MaxEnt Comparison
  - Lattice Rescoring
- Conclusion & Future work

Outline

- **Motivation**
- Sparse Non-negative Matrix Language Model
- Skip-grams
- Experiments, investigating:
  - Modeling Power (sentence level)
  - Computational Complexity
  - Cross-sentence Modeling
  - MaxEnt Comparison
  - Lattice Rescoring
- Conclusion & Future work

Motivation

- (Gated) Recurrent Neural Networks:
  - Current state of the art
  - Do not scale well to large data => slow to train/evaluate
- Maximum Entropy:
  - Can mix arbitrary features, extracted from large context windows
  - Log-linear model => suffers from same normalization issue as RNNLM
  - Gradient descent training for large, distributed models gets expensive
- Goal: build computationally efficient model that can mix arbitrary features (a la MaxEnt)
  - computationally efficient: O(counting relative frequencies)

Outline

- Motivation
- **Sparse Non-negative Matrix Language Model**
- Skip-grams
- Experiments, investigating:
  - Modeling Power (sentence level)
  - Computational Complexity
  - Cross-sentence Modeling
  - MaxEnt Comparison
  - Lattice Rescoring
- Conclusion & Future work

Sparse Non-Negative Language Model

- Linear Model: $$P(y|x) = \frac{\sum_t f_t(x, y)}{\sum_t \sum_{y'} f_t(x, y')}$$
- **Initialize** features with relative frequency: $$f_t^i(x, y) = count_t(x, y)/count_t(x)$$
- **Adjust** using exponential function of meta-features: $$f_t(x, y) = f_t^i(x, y)e^{\sum_m meta_m(t,x,y)}$$
    - Meta-features: template t, context x, target word y, feature $count_t(x, y)$, context count $count_t(x)$, etc + exponential/quadratic expansion
    - Hashed into 100K-100M parameter range
    - Pre-compute row sums => efficient model evaluation at inference time, proportional to number of active templates

Adjustment Model meta-features

- Features: can be anything extracted from (context, predicted word)
  - `[the quick brown fox]`
- Adjustment model uses *meta-features* to share weights e.g.
  - Context feature identity: `[the quick brown]`
  - Feature template type: 3-gram
  - Context feature count
  - Target word identity: `[fox]`
  - Target word count
  - Joins, e.g. context feature and target word count
- Model defined by the meta-feature weights and the feature-target relative frequency:

$$f_t(x, y) = f_t^i(x, y)e^{\sum_m meta_m(t,x,y)}$$

Parameter Estimation

- Stochastic Gradient Ascent on subset of training data
- Adagrad adaptive learning rate
- Gradient sums over entire vocabulary => use |V| binary predictors

- **Overfitting**: adjustment model should be trained on data disjoint with the data used for counting the relative frequencies
  - leave-one-out (here)
  - small held-out data (100k words) to estimate the adjustment model using multinomial loss
    - model adaptation to held-out data, see [Chelba and Pereira, 2016]

- More optimizations:
  - see paper for details, in particular efficient leave-one-out implementation

Outline

- Motivation
- Sparse Non-negative Matrix Language Model
- **Skip-grams**
- Experiments, investigating:
  - Modeling Power (sentence level)
  - Computational Complexity
  - Cross-sentence Modeling
  - MaxEnt Comparison
  - Lattice Rescoring
- Conclusion & Future work

Skip-grams

- Have been shown to compete with RNNLMs
- Characterized by tuple `(r,s,a)`:
  - `r` denotes the number of remote context words
  - `s` denotes the number of skipped words
  - `a` denotes the number of adjacent context words
- Optional tying of features with different values of s
- Additional `skip-</s>` features for cross-sentence experiments

Model	n	r	s	a	tied
SNM5-skip	1..5	1..3	1..3	1..4	no
		1..2	4..*	1..4	yes
SNM10-skip	1..10	1..(5-a)	1	1..(5-r)	no
		1	1..10	1..3	yes

Outline

- Motivation
- Sparse Non-negative Matrix Language Model
- Skip-grams
- **Experiments, investigating:**
    - Modeling Power (sentence level)
    - Computational Complexity
    - Cross-sentence Modeling
    - MaxEnt Comparison
    - Lattice Rescoring
- Conclusion & Future Work

Experiment 1: One Billion Word Benchmark

- Train data: ca. 0.8 billion tokens
- Test data: 159658 tokens
- Vocabulary: 793471 words
- OOV rate on test data: 0.28%
- OOV words mapped to <unk>, also part of vocabulary
- Sentence order randomized
- More details in [Chelba et al., 2014]

Model	Params	PPL
KN5	1.76 B	67.6
SNM5 (proposed)	1.74 B	70.8
SNM5-skip (proposed)	62 B	54.2
SNM10-skip (proposed)	33 B	52.9
RNNME-256	20 B	58.2
RNNME-512	20 B	54.6
RNNME-1024	20 B	51.3
SNM10-skip+RNNME-1024		41.3
ALL		41.0

TABLE 2: Comparison with all models in Chelba et al., 2014

Computational Complexity

- Complexity analysis: see paper
- Runtime comparison (in machine hours):

Model	Runtime
KN5	28h
SNM5	115h
SNM10-skip	487h
RNNME-1024	5760h

TABLE 3: Runtimes per model

Experiment 2: 44M Word Corpus

- Train data: 44M tokens
- Check data: 1.7M tokens
- Test data: 13.7M tokens
- Vocabulary: 56k words
- OOV rate:
  - check data: 0.89%
  - test data: 1.98% (out of domain, as it turns out)
- OOV words mapped to `<unk>`, also part of vocabulary
- Sentence order NOT randomized => allows cross-sentence experiments
- More details in [Tan et al., 2012]

Model	Check	Test
KN5	104.7	229.0
SNM5 (proposed)	108.3	232.3
SLM	-	279
n-gram/SLM	-	243
n-gram/PLSA	-	196
n-gram/SLM/PLSA	-	176
SNM5-skip (proposed)	89.5	198.4
SNM10-skip (proposed)	87.5	195.3
SNM5-skip-</s> (proposed)	79.5	176.0
SNM10-skip-</s> (proposed)	78.4	174.0
RNNME-512	70.8	136.7
RNNME-1024	68.0	133.3

TABLE 4: Comparison with models in [Tan et al., 2012]

Experiment 3: MaxEnt Comparison (Thanks Diamantino Caseiro!)

- Maximum Entropy implementation that uses hierarchical clustering of the vocabulary (HMaxEnt)
- Same hierarchical clustering used for SNM (HSNM)
  - Slightly higher number of params due to storing the normalization constant

Model	# params	PPL
SNM 5G	*1.7B*	*70.8*
KN 5G	*1.7B*	*67.6*
HMaxEnt 5G	2.1B	78.1
HSNM 5G	2.6B	67.4
HMaxEnt	5.4B	65.5
HSNM	6.4B	61.4

- One Billion Word benchmark:
  - HSNM perplexity is slightly better than HMaxEnt counterpart
- ASR exps on two production systems (Italian and Hebrew):
  - about same for dictation and voice search (+/- 0.1% abs WER)
  - SNM uses 4000X fewer resources for training (1 worker x 1h vs 500 workers x 8h)

Google

Outline

- Motivation
- Sparse Non-negative Matrix Language Model
- Skip-grams
- Experiments, investigating:
    - Modeling Power (sentence level)
    - Computational Complexity
    - Cross-sentence Modeling
    - MaxEnt Comparison
    - Lattice Rescoring
- **Conclusion & Future Work**

Conclusions & Future Work

- **Arbitrary categorical features**
  - same expressive power as Maximum Entropy
- **Computationally cheap**:
  - O(counting relative frequencies)
  - ~10x faster (machine hours) than specialized RNN LM implementation
  - easily parallelizable, resulting in much faster wall time
- **Competitive and complementary with RNN LMs**

Conclusions & Future Work

Lots of **unexplored potential**:

- Estimation:
  - replace the empty context (unigram) row of the model matrix with context-specific RNN/LSTM probabilities; adjust SNM on top of that
  - adjustment model is invariant to a constant shift: regularize
- Speech/voice search:
  - mix various data sources (corpus tag for skip-/n-gram features)
  - previous queries in session, geo-location, [Chelba and Shazeer, 2015]
  - discriminative LM: train adjustment model under N-best re-ranking loss
- Machine translation:
  - language model using window around a given position in the source sentence to extract conditional features f(target,source)

References

- Chelba, Mikolov, Schuster, Ge, Brants, Koehn and Robinson. One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling. In *Proc. Interspeech*, pp. 2635-2639, 2014.

- Chelba and Shazeer. Sparse Non-negative Matrix Language Modeling for Geo-annotated Query Session Data. In *Proc. ASRU*, pp. 8-14, 2015.

- Chelba and Pereira. Multinomial Loss on Held-out Data for the Sparse Non-negative Matrix Language Model. *arXiv:1511.01574*, 2016.

- Tan, Zhou, Zheng and Wang. A Scalable Distributed Syntactic, Semantic, and Lexical Language Model. *Computational Linguistics*, 38(3), pp. 631-671, 2012.