Using Dependency Grammar Features in Whole Sentence Maximum Entropy Language Model for Speech Recognition

Teemu Ruokolainen, Tanel Alumäe, Marcus Dobrinkat

October 8th, 2010

Contents

Whole sentence language modeling

Statistical sentence modeling problem

- Given a finite set of observed sentences, learn a model which gives useful probability estimates for arbitrary new sentences

n-gram model: the standard approach

- Model language as a high-order Markov Chain; current word is dependent only on $n - 1$ of its preceeding words
- Sentence probability is obtained using chain rule; sentence probability is product of word probabilities
- Modeling is based on local dependencies of the language only; grammatical regularities learned by the model will be captured implicitly within the short word windows

Example: n-gram succeeds

- Stock markets **fell** yesterday.
  - Log probability given by trigram LM = -19.39
- Stock markets **fallen** yesterday.
  - Log probability = -21.26

Example: n-gram fails

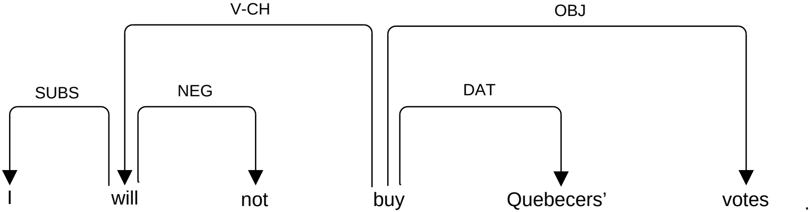
- Stocks **have** by and large **fallen**.
  - Log probability = -19.92
- Stocks **have** by and large **fell**.
  - Log probability = -18.82

Our aim

- Explicit modeling of grammatical knowledge over whole sentence
  - Dependency Grammar Features
  - Whole Sentence Maximum Entropy Language Model (WSME LM)
  - Experiments in a large vocabulary speech recognition task

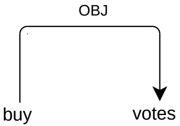
Dependency Grammar

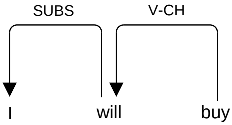
- Dependency parsing results in head-modifier relations between pairs of words, together with the labels of the relationships
- The labels describe the type of the relation, e.g. subject, object, negate
- These asymmetric bilexical relations define a complete dependency structure for the sentence



Extracting Dependency Grammar Features

- Dependencies are converted into binary features
  - Feature is or is not present in a sentence
- Dependency bigram features contain a relationship between a head and a modifier
- Dependency trigram features contain a modifier with its head and the head's head





Whole Sentence Maximum Entropy Language Model (WSME LM)

Principle of Maximum Entropy

- Model selection criterion
- From all the probability distributions satisfying known constraints, choose the one with the highest entropy

Maximum Entropy Model

- Constraints: expected values of features
- Form of the model satisfying the constraints: exponential distribution
- Within the exponential model family: maximum likelihood solution is the maximum entropy solution

WSME LM

- WSME LM is the exponential probability distribution over sentences which is closest to the background n-gram model (in Kullback-Leibler divergence sense) while satisfying linear constraints specified by empirical expectations of features
  - For uniform background model, the Maximum Entropy solution
- For testing data, the sentence probabilities given by the n-gram model are, effectively, scaled according to the features present in the sentence.

Practical issues

- Training WSME LM requires sentence samples from the exponential model
  - Markov Chain Monte Carlo sampling methods

Experiments

Experiment setup

- Train a baseline n-gram LM and WSME LM
- Obtain an N-best hypothesis list for a sentence from speech recognizer using the baseline n-gram and rescore them using WSME LM
- Compare model performance with speech transcript perplexity and speech recognition word error rate (WER)

Data

- Textual training corpus: Gigaword
  - English newswire articles of typical daily news topics; sports, politics, finances, etc.
  - 1M sentences (20M words)
  - Small subset of Gigaword
- Speech test corpus: Wall Street Journal
  - Dictated English financial newswire articles
  - 329 sentences (11K words)

Baseline LM

- Trigram model trained using Kneser-Ney smoothing
- Vocabulary size: 60K words

Dependency parsing

- Textual data was parsed using a freely distributed Connexor Machine Syntax parser

WSME LM training

- Sentence samples from the exponential model were obtained using importance sampling
- The L-BGFS algorithm was used for optimizing the parameters
- The parameters of the model were smoothed using Gaussian priors

Speech recognition system

- Large vocabulary speech recognizer developed at the Department of Information and Computer Science, Aalto University

Experiment results

- We observe a 19% relative decline in perplexity (PPL) when using the WSME LM compared to baseline trigram
- The WER drops by 6.1% relative (1.8% absolute) compared to the baseline
- Note: Results reported only for trigram Dependency Grammar features
- Performance gain is significant

Table: Perplexity (PPL) and word error rate (WER) when using different language models.

Language model	PPL	WER
Word trigram	303	29.6
WSME LM	**244**	30.6
Word trigram + WSME LM	255	**27.9**

Conclusions

- We described our experiments with WSME LM using binary features extracted with a dependency grammar parser
- The dependency features were in the form of labeled asymmetric bilexical relations
  - Experiments on bigram and trigram features
- The WSME LM was evaluated in a large vocabulary speech recognition

Conclusions (continued)

- We obtained significant improvement in performance using WSMELM compared to a baseline word trigram
- WSME LMs provide an elegant way to combine statistical models with linguistic information
- The main shortcoming of the method; extremely high memory consumption requirement during training of the model