Topic-Based Language Models Using EM

Teemu Hirsimäki

31.10.2001

Language models

- Language models assign probabilities to word sequences.
- n-gram models have been widely used.
  - They are simple and surprisingly effective.
  - However, they ignore long-range dependencies.
- Topic based models try to model long-range dependencies.
  - The order of the word history is often ignored.
  - Combination with n-gram models is often necessary.
  - Computationally heavy.

Topic decomposition by EM

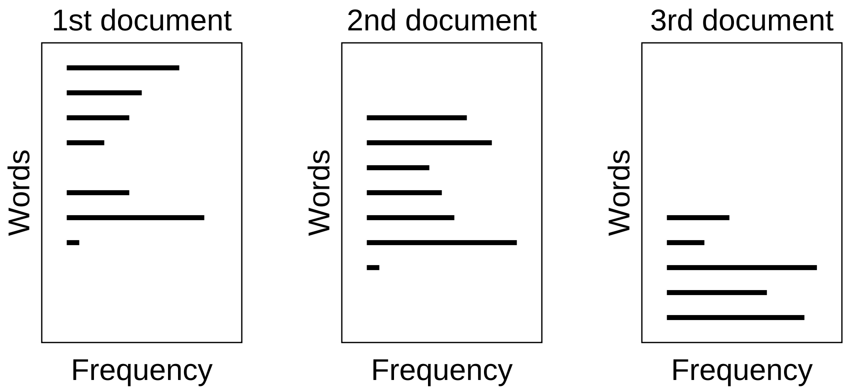
- The model is a composition of topic factors:

$$P(w|h) = \sum_t P(w|t)P(t|h).$$

- $P(w|t)$ are topic-specific word probabilities, estimated in training phase.
- $P(t|h)$ are history-based mixing proportions, estimated in test phase.

Estimation of the word probabilities

- A collection of documents is used to estimate $p(w|t)$.
- Each document is presented as a word vector.



- Matrix $n(w,d)$ over a fixed vocabulary.
- In estimation process we also need a probability distribution of topics for each document, $p(t|d)$.
- Random initialization for $p(w|d)$ and $p(t|d)$.
- The number of topics is fixed.

Estimation of the word probabilities by EM

- *E-step*: For each word in each document, we calculate the probability that the word was generated by the topic factor t:

$$P(t|w,d) \propto P(w|t)P(t|d). \tag{1}$$

- *M-step*: The model parameters are adjusted as follows:

$$P(w|t) \propto \sum_d n(w,d)P(t|w,d) \tag{2}$$

$$P(t|d) \propto \sum_w n(w,d)P(t|w,d). \tag{3}$$

Using the model for testing

- The original model was $P(w|h) = \sum_t P(w|t)P(t|h)$.
- $P(w|t)$ is estimated already, but $P(t|h)$ is still needed.
- $P(t|h)$ could be estimated iterating (1) and (3), but an online approximation was presented in the paper.
- Using full EM iterations showed only neglible imrpovements.

Experiments on TDT-1 corpus

- TDT-1 corpus of newspaper text and transcribed broadcast news stories.
- 6,797,659 million words in 15,862 documents.
- Vocabulary consisted of all words occurring twice: 49,225 words. (Also from test data?)
- 90% for training, 10% for testing.
- 256 topic factors.

Model	Perplexity
Unigram	1140.6
Topic model	829.1
Trigram	205.2
Linear interpolation: $\lambda P_{tri} + (1-\lambda) P_{topic}$	189.2
Log-scale interpolation: $P_{tri}^{\lambda} * P_{topic}^{1-\lambda}$	180.8
Unigram rescaling: $P_{tri} * \frac{P_{topic}}{P_{unigram}}$	170.1

Notes on the TDT-1 experiment

- 10 stories, 4274 words (quite small test set!)
- They ran another test also on 50 stories with 24,850 words. The results were: 180.8 for trigram, and 147.2 for unigram scaling.
- The paper did not say anything about out-of-vocabulary words.
- Normalizing over vocabulary makes the method quite heavy: 10 words per minute on a 296 *MHz* Ultrasparc.

When does the model help?

- There are very common function words such as “and”, “of”, and “the”, which occur in all documents. The topic model does not predict these words well.
- The topic model was worse than a trigram for about the 100 most frequent words.
- Improvement: use the n-gram probability the for the most frequent words, and combined model for the other words. This leads to computational savings.
- Also the length of history affects the performance of the topic model: the combined model outperformed the trigram model on histories longer than 5 words. Increasing the history size over 19 words did not help.

Speech recognition experiments

- The topic model was trained on large corpus: 60,328,305 words in 124,814 documents.
- However, the topic model did not decrease the recognition error rate: 35.6% using trigram, and 36.5% using the combined model.
- The topic model tends to improve on longer content words, which are already recognized relatively well.

Conclusions

- A probabilistic topic-based model can be estimated using the EM technique.
- Combined with trigram model, the topic model seems to improve perplexity results considerably.
- The topic model seems to help most on rare words with a history of 5–19 words.
- However, the model is computationally heavy, and does not seem to help in speech recognition.