Reading Group Presentation Report from 26.03.2012

Paper Distributed Word Clustering for Large Scale Class-Based Language Modeling in Machine Translation by Jakob Uszkoreit and Thorsten Brants

Presented by Long DT

Report by Joachim Daiber

1 Overview and Notes from the Paper

1.1 Difference between predictive and two-side class-based model

Two-side class based model:

$$P(w_i|w_1^{i-1}) \approx p_0(w_i|c(w_i))p_1(c(w_i)|c(w_{i-n+1}^{i-1}))$$

Predictive class-based model:

$$P(w_i|w_1^{i-1}) \approx p_0(w_i|c(w_i))p_1(c(w_i)|w_{i-n+1}^{i-1})$$

The main difference is the use of words instead of classes for the history of p_1.

Why does this improve the algorithm?

- improves complexity, easier to distribute
- after a discussion in class which lead to no conclusion, we assumed that the choice was empirical

1.2 Exchange clustering

- why do we want to use classes at all? sparseness of the data!
- move word from one cluster to another
- recompute perplexity
- choose exchange that maximizes perplexity
- Can one word be assigned to multiple classes? No, this would be fuzzy clustering.
- too complicated! Improve: recalculate step-by-step, maintain array (dynamic programming)

1.3 Complexity of Exchange Clustering

$$O(I \cdot (2 \cdot B + N_v \cdot N_c \cdot (N_c^{pre} + N_c^{suc})))$$

- $N_c^{pre} + N_c^{suc}$ are the average number of clusters preceding and succeeding another cluster
- B is the number of distinct bigrams. We maintain an array (dynamic programming). In the formula, we have $2 \cdot B$ because as the cost of maintaining this array (2 because of previous word and next word).
- I is the number of iterations

1.4 Predicitive Exchange Clustering

$$P(w_i|w_1^{i-1}) \approx p_0(w_i|c(w_i)) \cdot p_1(c(w_i|w_{i-1})) = \frac{N(w_i)}{N(c(w_i))} \cdot \frac{N(w_{i-1}, c(w_i)}{N(w_{i-1})}$$

- equations (6)-(10) in the paper demonstrate the new way to calculate the perplexity: when moving a word from c to c', last part of (10) ($-\sum_{c \in C} N(c) \cdot \log N(c)$) must be recalculated, the first part ($\sum_{v \in V, c \in suc(v)} N(v, c) \cdot \log N(v, c)$) can be quickly calculated with an additional array

1.5 New complexity

$$O(I \cdot N_c \cdot (B + N_v))$$

- B is the number of distinct bigrams
- I is the number of iterations
- N_c is the number of clusters
- N_v is the size of the vocabulary

The advantage: only two classes affected by a move of a word from one class to another.

1.6 Distributed clustering

- divide vocabulary into subsets
- each subset is given to one worker
- each worker has counts from the previous iteration, workers must synchronize after each iteration

1.7 Experiments

- experiments run using the LM in phrase-based machine translation setup
- computed BLEU score with different LMs (word-based only and class-based with different numbers of clusters, see table (1) in the paper)
- computed BLEU scores for Arabic-English translation with 5-gram predictive class-based model with different inputs (see table (2))
- computed BLEU scores for English-Arabic translation with 5-gram predictive class-based model with different inputs (see table (3))

1.8 Conclusion

- the changes to the algorithm show big performance improvements: there is an improvement in complexity and the model can be used in distributed setting
- the model improves quality of state-of-the art machine translation

2 Questions discussed in the Reading Group

- Do the reported improved results use a model that includes an only word-based model?
  - Yes, class-based and word-based model are always combined (two parts in a log-linear model)
  - improvement could be to merge the two models in the LM itself, because whether or not to use the class-based model may depend on the history
- How exactly does the method distribute data/computations to workers?
  - before first iteration
    * sort words, assign to clusters
    * compute counts from clustering
  - distribute vocabulary to workers (1/10 each)
    * map: each worker: take out 1/3 (this is an empirical choice! It may not converge when this value is > 1/2) of the vocabulary, compute updates for it
    * there is a tradeoff between the number of iterations and the size of the data you give the workers
      · more iterations: may need to wait for workers, there may be overhead in initializing the workers
    * reduce: combine difference in counts from workers, sum up, map again

3 Checkpoint questions

- What is the predictive class based model?
  - similar to two-side class based model but with word instead of class in history of p1, see above
- Why do we only consider clusterings for which N(w, c) > 0 or N(c, w) > 0 in the exchange algorithm?
  - because only the clusters that satisfy this condition are affected
- What's better in predictive exchange clustering?
  - observation: better results for some data sets
  - better complexity
  - easier to distribute