Info 2950, Lecture 21

20 Apr 2017

Prob Set 6: due Mon night 24 Apr

Prob Set 7: due Tue night 2 May (?)

Prob Set 8: due Wed night 10 May (?)

More Data

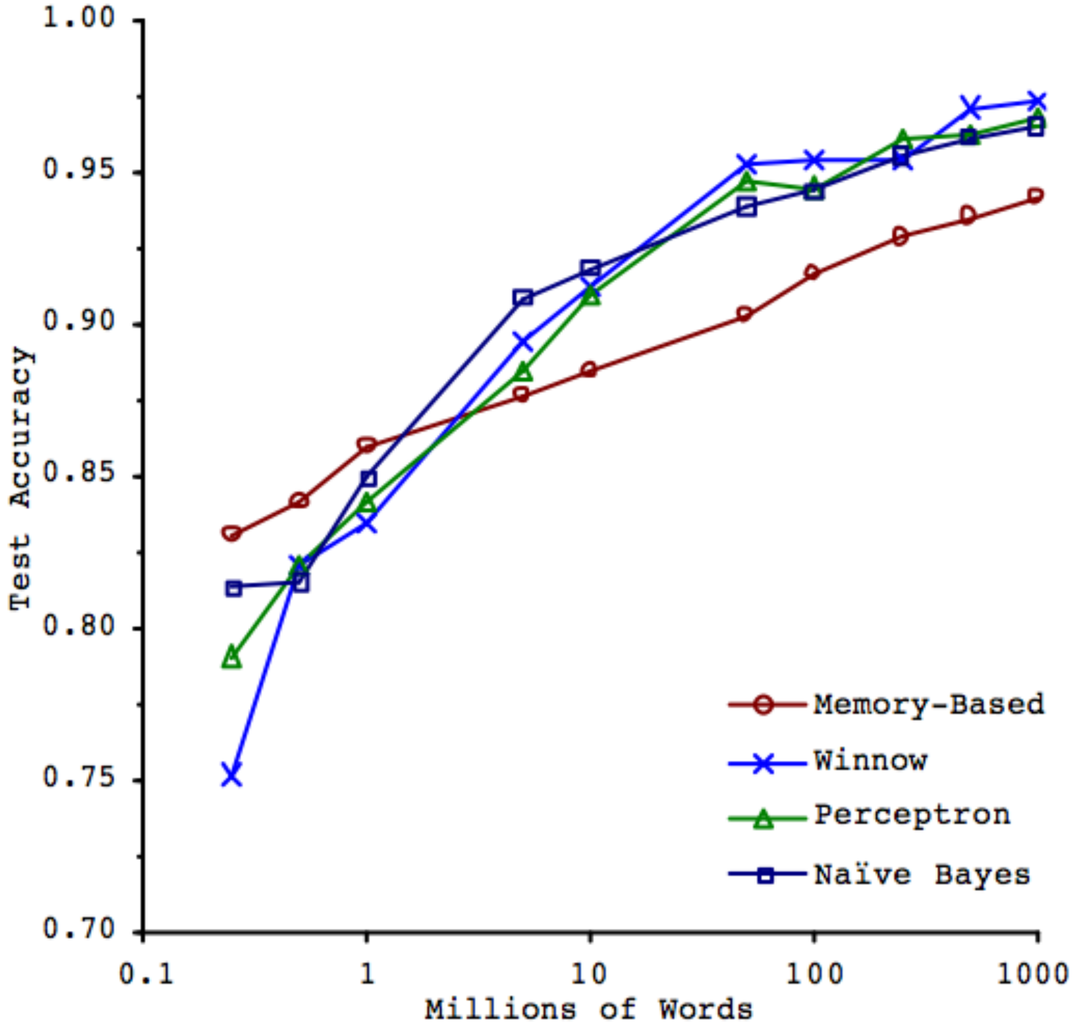


Figure 1. Learning Curves for Confusion Set Disambiguation

http://acl.ldc.upenn.edu/P/P01/P01-1005.pdf

Scaling to Very Very Large Corpora for Natural Language Disambiguation

M. Banko and E. Brill (2001)

More Data for this Task

http://acl.ldc.upenn.edu/P/P01/P01-1005.pdf
Scaling to Very Very Large Corpora for Natural Language Disambiguation
M. Banko and E. Brill (2001)

The amount of readily available on-line text has reached hundreds of billions of words and continues to grow. Yet for most core natural language tasks, algorithms continue to be optimized, tested and compared after training on corpora consisting of only one million words or less. In this paper, we evaluate the performance of different learning methods on a prototypical natural language disambiguation task, confusion set disambiguation, when trained on orders of magnitude more labeled data than has previously been used. We are fortunate that for this particular application, correctly labeled training data is free. Since this will often not be the case, we examine methods for effectively exploiting very large corpora when labeled data comes at a cost.

(Confusion set disambiguation is the problem of choosing the correct use of a word, given a set of words with which it is commonly confused. Example confusion sets include: {principle , principal}, {then , than}, {to , two , too} , and {weather,whether}.)

Segmentation

- nowisthetimeforallgoodmentocometothe
- Probability of a segmentation = P(first word) $\times$ P(rest)
- Best segmentation = one with highest probability
- P(word) = estimated by counting

Trained on 1.7B words English, 98% word accuracy

back to segmentation

e.g., unigram model for segmentation:

$$P(w_1 \dots w_n) = P(w_1) \dots P(w_n)$$

To segment ‘wheninrome’, consider candidates such as “when in rome”, and compute P(when) $\times$ P(in) $\times$ P(rome).
Gives best answer If product is larger than any other candidate’s.

’wheninthecourseofhumaneventsitbecomesnecessary’ has 35 trillion segmentations, but can be read by finding probable words in sequence (not by considering all 2^{n-1} segmentations)

So use the largest product recursively: P(first) $\times$ P(remaining)

Spelling with Statistical Learning

- Probability of a spelling correction, c = P(c as a word) × P(original is a typo for c)
- Best correction = one with highest probability
- P(c as a word) = estimated by counting
- P(original is a typo for c) = proportional to number of changes

Similarly for speech recognition, using language model $p(c)$ and acoustic model $p(s|c)$
(Russel & Norvig, "Artificial Intelligence", section 24.7)

And others

- Statistical Machine Translation
  - Collect parallel texts ("Rosetta stones"), Align (Brants, Popat, Xu, Och, Dean (2007), "Large Language Models in Machine Translation")
- fill in occluded portions of photos (Hayes and Efros, 2007)

Statistical Machine Translation

Google *n*-gram corpus created by researchers in the machine translation group (released 2006).
Translating from foreign language (f) into English (e) similar to correcting misspelled words.
The best English translation is modeled as:

$$\text{best} = \text{argmax}_e P(e|f) = \text{argmax}_e P(f|e)P(e)$$

where $P(e)$ is the language model for English, which is estimated by the word *n*-gram data, and $P(f|e)$ is the translation model, learned from a bilingual corpus (where pairs of documents are marked as translations of each other). Although top systems make use of many linguistic features, including parts of speech and syntactic parses of the sentences, seems that majority of knowledge necessary for translation resides in the *n*-gram data.

Further details in Brants,Popat,Xu,Och,Dean (2007)
"Large Language Models in Machine Translation",
http://acl.ldc.upenn.edu/D/D07/D07-1090.pdf

https://careers.google.com/stories/how-one-team-turned-the-dream-of-speech-recognition-into-a-reality/

Other Tasks

- Secret codes
- Language Identification
- Spam Detection and Other Classification Tasks
- Author Identification (Stylometry)

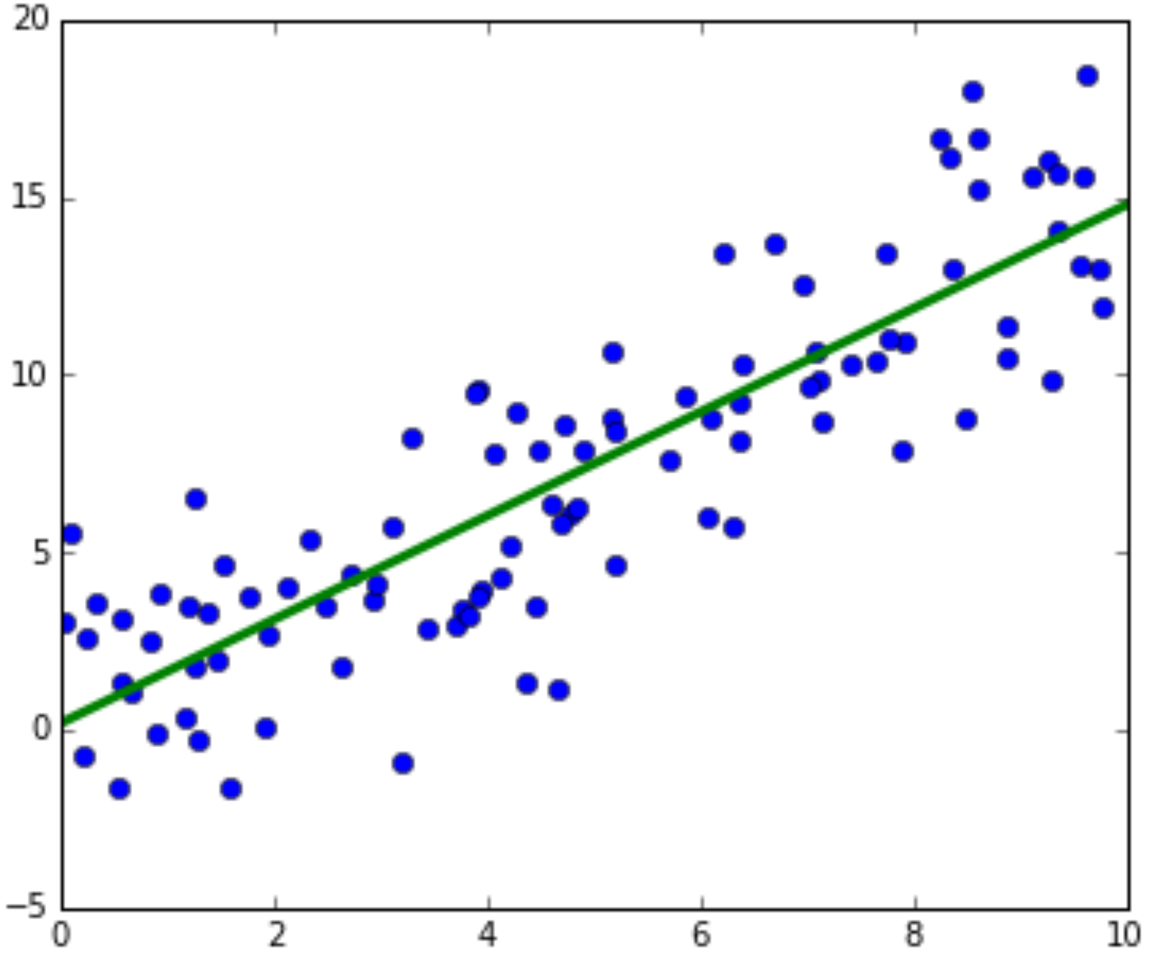
General "Big Data" Procedure

- Define a probabilistic model
  (i.e., use data to create language model, a probability distribution over all strings in the language, learned from corpus, and use model to determine probability of candidates)
- Enumerate candidates
  (e.g., segmentations, corrected spellings)
- Choose the most probable candidate:

$$\text{best} = \text{argmax}_{c \in \text{candidates}} P(c)$$

Python: best = max(candidates, key=P)

Big Data = Simple Algorithm


20
15
10
5
0
−5
0
2
4
6
8
10