Search Engine Architecture

4. Modeling and Evaluation

Noted slides adapted from: Lin et al.'s Big Data Infrastructure, UMD Spring 2015 with cosmetic changes.

Today's Agenda

- Language models
  - Application to statistical translation
- Retrieval models
  - Preprocessing
  - Scoring
- Model evaluation

Language Models

Language Models

$$P(w_1, w_2, \ldots, w_T)$$

$$= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \ldots P(w_T|w_1, \ldots, w_{T-1})$$

[chain rule]

Is this tractable?

Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

When estimating distributions…

- Two important rules
  - Probabilities must sum to one
  - Smooth

Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

Approximating Probabilities

Basic idea: limit history to fixed number of words *N* (Markov Assumption)

$$P(w_k|w_1,\ldots,w_{k-1}) \approx P(w_k|w_{k-N+1},\ldots,w_{k-1})$$

N=1: Unigram Language Model

$$P(w_k|w_1,\ldots,w_{k-1}) \approx P(w_k)$$

$$\Rightarrow P(w_1,w_2,\ldots,w_T) \approx P(w_1)P(w_2)\ldots P(w_T)$$

Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

Approximating Probabilities

Basic idea: limit history to fixed number of words *N* (Markov Assumption)

$$P(w_k|w_1,\ldots,w_{k-1}) \approx P(w_k|w_{k-N+1},\ldots,w_{k-1})$$

N=2: Bigram Language Model

$$P(w_k|w_1,\ldots,w_{k-1}) \approx P(w_k|w_{k-1})$$

$$\Rightarrow P(w_1,w_2,\ldots,w_T) \approx P(w_1|< \mathrm{S} >)P(w_2|w_1)\ldots P(w_T|w_{T-1})$$

Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

Approximating Probabilities

Basic idea: limit history to fixed number of words *N* (Markov Assumption)

$$P(w_k|w_1,\ldots,w_{k-1}) \approx P(w_k|w_{k-N+1},\ldots,w_{k-1})$$

N=3: Trigram Language Model

$$P(w_k|w_1,\ldots,w_{k-1}) \approx P(w_k|w_{k-2},w_{k-1})$$

$$\Rightarrow P(w_1,w_2,\ldots,w_T) \approx P(w_1|<\mathrm{S}><\mathrm{S}>)\ldots P(w_T|w_{T-2}w_{T-1})$$

Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

Building *N*-Gram Language Models

- Compute maximum likelihood estimates (MLE) for individual *n*-gram probabilities

  - Unigram: $P(w_i) = \frac{C(w_i)}{N}$

  - Bigram: $P(w_i, w_j) = \frac{C(w_i, w_j)}{N}$

    $$P(w_j|w_i) = \frac{P(w_i, w_j)}{P(w_i)} = \frac{C(w_i, w_j)}{\sum_w C(w_i, w)} = \frac{C(w_i, w_j)}{C(w_i)}$$

  - Generalizes to higher-order *n*-grams

Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

Thou shalt smooth!

- Zeros are bad for any statistical estimator
  - Need better estimators because MLEs give us a lot of zeros
  - A distribution without zeros is "smoother"
- The Robin Hood Philosophy: Take from the rich (seen *n*-grams) and give to the poor (unseen *n*-grams)
  - And thus also called discounting
  - Make sure you still have a valid probability distribution!
- Lots of techniques:
  - Laplace, Good-Turing, Katz backoff, Jelinek-Mercer
  - Kneser-Ney represents best practice

Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

Stupid Backoff

- Let's break all the rules:

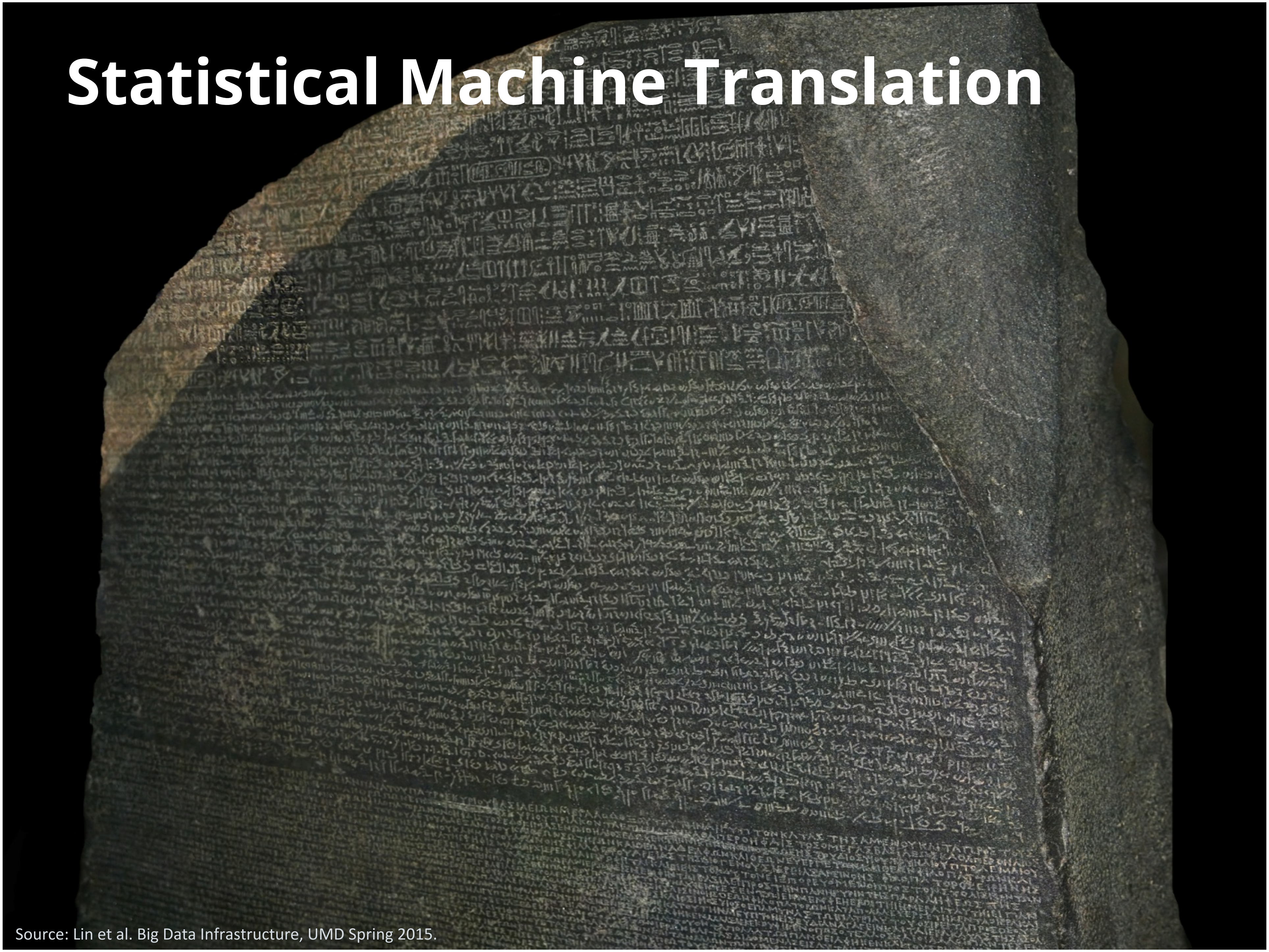
$$S(w_i|w_{i-k+1}^{i-1}) = \begin{cases} \frac{f(w_{i-k+1}^{i})}{f(w_{i-k+1}^{i-1})} & \text{if } f(w_{i-k+1}^{i}) > 0 \\ \alpha S(w_i|w_{i-k+2}^{i-1}) & \text{otherwise} \end{cases}$$

$$S(w_i) = \frac{f(w_i)}{N}$$

- But throw *lots* of data at the problem!

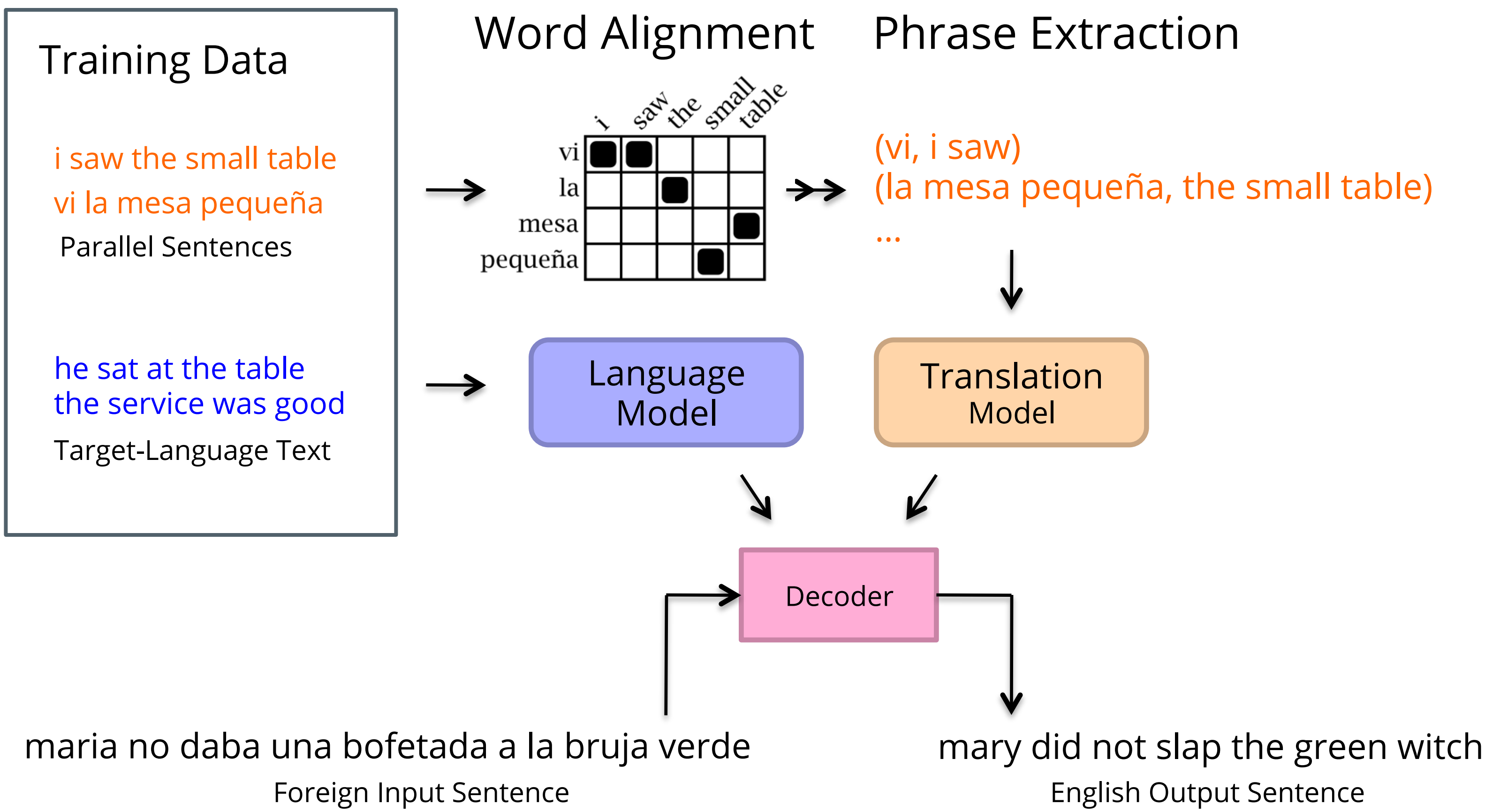
Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

Statistical Machine Translation



Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

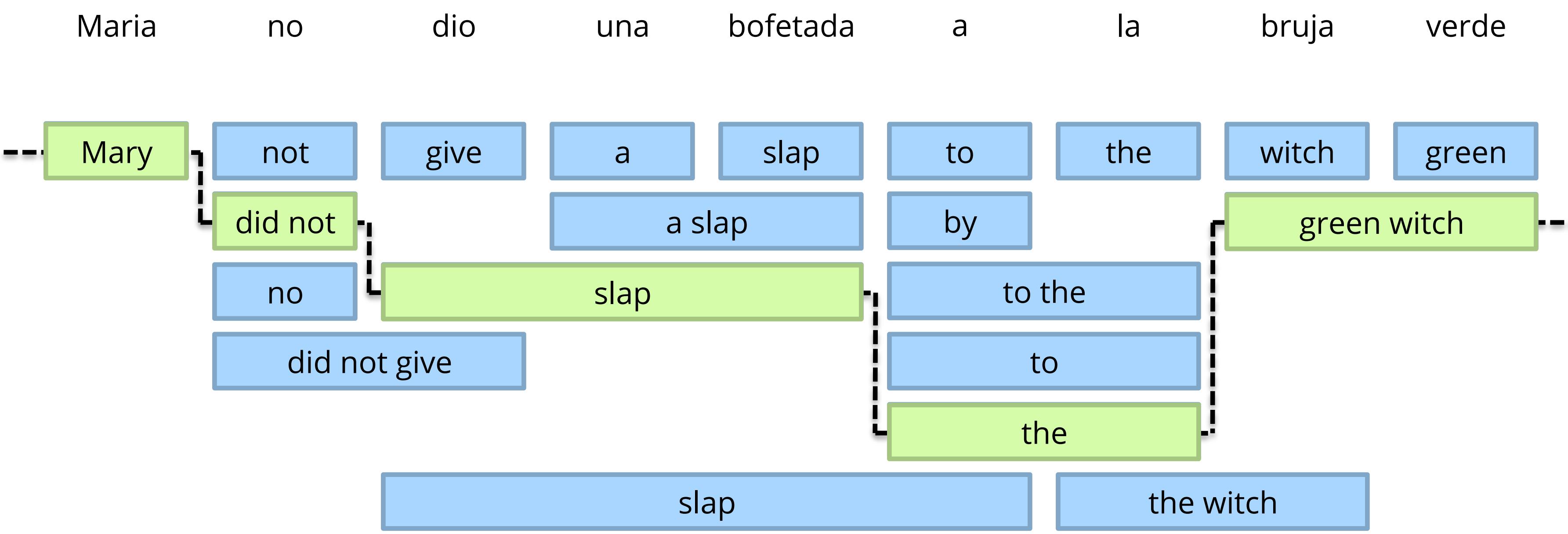
Statistical Machine Translation



$$\hat{e}_1^I = \arg\max_{e_1^I}\left[P(e_1^I \mid f_1^J)\right] = \arg\max_{e_1^I}\left[P(e_1^I)P(f_1^J \mid e_1^I)\right]$$

Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

Translation as a Tiling Problem



$$\hat{e}_1^I = \arg\max_{e_1^I}\left[P(e_1^I \mid f_1^J)\right] = \arg\max_{e_1^I}\left[P(e_1^I)P(f_1^J \mid e_1^I)\right]$$

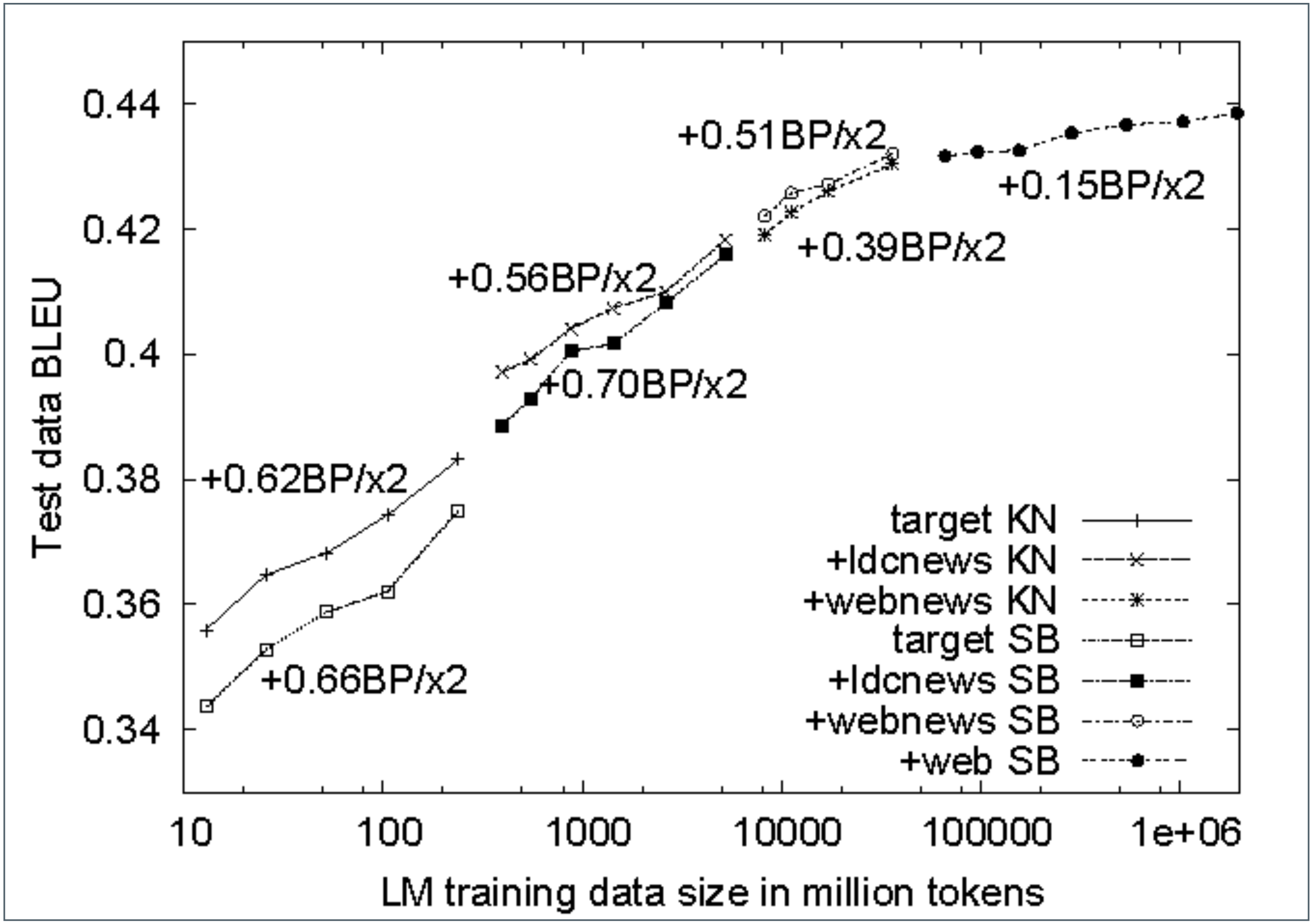
Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

Results: Running Time

	target	*webnews*	*web*
# tokens	237M	31G	1.8T
vocab size	200k	5M	16M
# n-grams	257M	21G	300G
LM size (SB)	2G	89G	1.8T
time (SB)	20 min	8 hours	1 day
time (KN)	2.5 hours	2 days	–
# machines	100	400	1500

Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

Results: Translation Quality



Source: Brants et al. (EMNLP 2007) via Data Intensive Computing with MapReduce 2013

Retrieval Models

Today's Agenda

- Language models
  - Application to statistical translation
- Retrieval models
  - Preprocessing
  - Scoring
- Model evaluation

How do we represent text?

- Remember: computers don't "understand" anything!
- "Bag of words"
  - Treat all the words in a document as index terms
  - Assign a "weight" to each term based on "importance" (or, in simplest case, presence/absence of word)
  - Disregard order, structure, meaning, etc. of the words
  - Simple, yet effective!
- Assumptions
  - Term occurrence is independent
  - Document relevance is independent
  - "Words" are well-defined

Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

What’s a word?

天主教教宗若望保祿二世因感冒再度住進醫院。這是他今年第二度因同樣的病因住院。

وقال مارك ريجيف - الناطق باسم الخارجية الإسرائيلية - إن شارون قبل الدعوة وسيقوم للمرة الأولى بزيارة تونس، التي كانت لفترة طويلة المقر الرسمي لمنظمة التحرير الفلسطينية بعد خروجها من لبنان عام 1982.

Выступая в Мещанском суде Москвы экс-глава ЮКОСа заявил не совершал ничего противозаконного, в чем обвиняет его генпрокуратура России.

भारत सरकार ने आर्थिक सर्वेक्षण में वित्तीय वर्ष 2005-06 में सात फ़ीसदी विकास दर हासिल करने का आकलन किया है और कर सुधार पर ज़ोर दिया है

日米連合で台頭中国に対処…アーミテージ前副長官提言

조재영 기자= 서울시는 25일 이명박 시장이 `행정중심복합도시'' 건설안에 대해 `군대라도 동원해 막고싶은 심정''이라고 말했다는 일부 언론의 보도를 부인했다.

Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

Preprocessing

- Case folding
- Tokenization
- Stop words
- Stemming
- Collocations

Case Folding

- Convert all terms to lower case
  - Users will use lower case in queries anyway
- What about proper nouns? Acronyms?
  - Exception: upper case in middle of sentence?
- Retaining case information might be useful for other features
  - E.g. recognizing named entities

Tokenization

- Input: “To be, or not to be”
- Output: to, be, or, not, to, be
- Issues:
  - “New York University”
  - “Shakespeare’s play”
  - “state-of-the-art”

Stopwords

- A, an, to, of, ...
- Issues:
  - What if you query for a phrase?
- Other ways to reduce importance of common terms...

Stemming

- Heuristic-based removal of prefixes and suffixes
- E.g. Porter stemmer
  - “sses” -> “ss” (caresses -> caress)
  - “ies” -> “I” (ponies -> poni)
  - “s” -> “” (cats -> cat)
- Porter, Snowball, Lancaster stemmers
  - In increasing likelihood of overstemming
- (Cf. lemmatization)

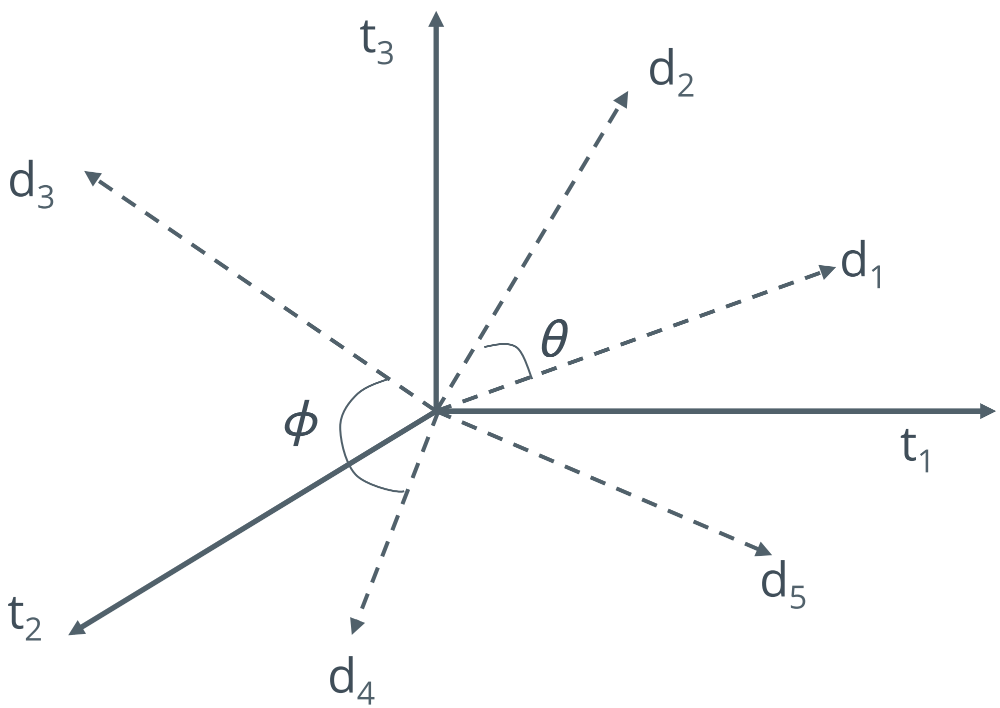
Collocations

- Find phrases for use downstream
- Student's t
- Chi square
- Pointwise mutual information
- Likelihood ratio
- Variations of $p(x, y) / (p(x) + p(y))$

Retrieval

- TF-IDF
- Variants
- Norms
- Coordination
- Boosting

Vector Space Model



Assumption: Documents that are “close together” in vector space “talk about” the same things

Therefore, retrieve documents based on how close the document is to the query (i.e., similarity ~ “closeness”)

Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

Similarity Metric

- Use "angle" between the vectors:

$$d_j = [w_{j,1}, w_{j,2}, w_{j,3}, \ldots w_{j,n}]$$
$$d_k = [w_{k,1}, w_{k,2}, w_{k,3}, \ldots w_{k,n}]$$

$$\cos\theta = \frac{d_j \cdot d_k}{|d_j||d_k|}$$

$$\mathrm{sim}(d_j, d_k) = \frac{d_j \cdot d_k}{|d_j||d_k|} = \frac{\sum_{i=0}^{n} w_{j,i} w_{k,i}}{\sqrt{\sum_{i=0}^{n} w_{j,i}^2}\sqrt{\sum_{i=0}^{n} w_{k,i}^2}}$$

- Or, more generally, inner products:

$$\mathrm{sim}(d_j, d_k) = d_j \cdot d_k = \sum_{i=0}^{n} w_{j,i} w_{k,i}$$

Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

Term Weighting

- Term weights consist of two components
  - Local: how important is the term in this document?
  - Global: how important is the term in the collection?
- Here's the intuition:
  - Terms that appear often in a document should get high weights
  - Terms that appear in many documents should get low weights
- How do we capture this mathematically?
  - Term frequency (local)
  - Inverse document frequency (global)

Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

TF-IDF Term Weighting

$$w_{i,j} = \mathrm{tf}_{i,j} \cdot \log \frac{N}{n_i}$$

$w_{i,j}$ weight assigned to term *i* in document *j*

$\mathrm{tf}_{i,j}$ number of occurrence of term *i* in document *j*

N number of documents in entire collection

n_i number of documents with term *i*

Variant – use sublinear tf (e.g. log tf)

Source: Lin et al. Big Data Infrastructure, UMD Spring 2015.

Normalization

- Divide vectors by some value
- L2 – square root of sum of squares
- L1 – sum of absolute values
- Tradeoffs but L2 more common

Coordination Factor

- Consider query “quick brown fox”
- One document contains “quick brown” many times
- Another contains “quick brown fox” only a few times
- Coord factor rewards occurrence of all three terms

Boosting

- Consider query “quick brown fox”
- One document contains “quick brown fox” in the title
- Another contains “quick brown fox” twice in the body
- Boosting reflects that first document is likely more relevant

Today's Agenda

- Language models
  - Application to statistical translation
- Retrieval models
  - Preprocessing
  - Scoring
- Model evaluation

Model Evaluation

Evaluation

- Which features are effective?
- Stop lists, stemming, IDF...
- Requires gold standard or ground truth
  - Standard test collections: TREC, Reuters, etc.
  - Click logs
- UI concerns as well

Why do we care?

- *Lean Startup* by Eric Ries:
  - Build
  - ***Measure***
  - Learn
- Biggest risk to a new venture isn't falling behind schedule or buggy software
  - It's building the wrong thing entirely!

Unranked Evaluation

- Simple case: search engine returns a set of results
- This is the case for Boolean retrieval
- Examples:
    - Precision
    - Recall
    - Accuracy?
    - F-Measure

Precision

- P(relevant | retrieved)

$$\text{precision} = \frac{\#\text{ relevant items retrieved}}{\#\text{ retrieved items}}$$

Recall

- P(retrieved | relevant)

$$\text{recall} = \frac{\#\ \text{relevant items retrieved}}{\#\ \text{relevant items}}$$

Accuracy

- Almost never a good idea
- If 10 in 10,000 documents are relevant, then never returning anything gives 99.9% accuracy
  - But recall is 0%

F Measure

- Harmonic mean of precision and recall

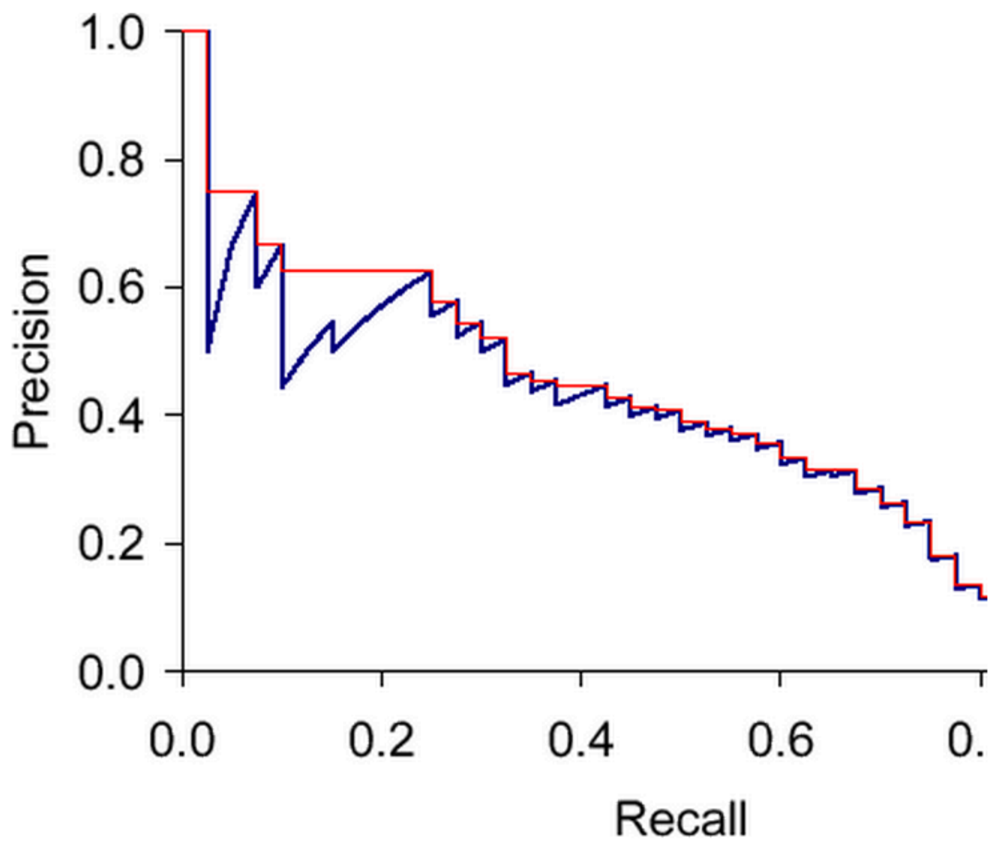
$$F = \frac{2PR}{P + R}$$

- Why harmonic mean?
  - If we use arithmetic mean, returning everything gives 100% recall, and 50% arithmetic mean
  - But if 1 in 10,000 documents is relevant then F=0.02%

Ranked Evaluation

- How do we evaluate effectiveness when top results should be more relevant?
  - Set measures on top K
  - Mean Average Precision
  - Cumulative Gain
  - NDCG

Precision/Recall Graph



- Blue line: P/R for increasing k
- Red line: highest P for a given R

Normalized Discounted Cumulative Gain

- CG is the sum of graded relevances:

$$\mathrm{CG_p} = \sum_{i=1}^{p} rel_i$$

- DCG penalizes relevance by position:

$$\mathrm{DCG_p} = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2(i)}$$

- nDCG normalizes to interval [0, 1]

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p}$$

Questions?