Statistical Language Models With Embedded Latent Semantic Knowledge

Jerome R. Bellegarda
Apple Computer, Inc.

Presented by Wen-Hung Tsai
NTNU CSIE

CONTENTS

Introduction

- The Bayesian approach pervasive in today's speech recognition systems entails the construction of a prior model of the language, as pertains to the domain of interest. The role of this prior, in essence, is to quantify which word sequences are acceptable in a given language for a given task, and which are not. It must therefore encapsulate as much as possible of the syntactic, semantic, and pragmatic characteristics of the domain.

Introduction

- In the past two decades, it has become increasingly common to do so through statistical *n*-gram language modeling (LM)
- Although widespread, this solution is not without drawbacks:
  - Prominent among the challenges faced by *n*-gram modeling is the inherent locality of its scope, due to the limited amount of context available for predicting each word

Scope Locality

- Central to this issue is the choice of *n*, which has implications in terms of predictive power and parameter reliability.
- Consider two equivalent phrases:
  stocks fell sharply as a result of the announcement (9.1)
  stocks, as a result of the announcement, sharply fell (9.2)
  the problem of predicting the word "*fell*" from the word "*stocks*"

Scope Locality

- In (9.1), this can be done with the help of a bigram LM ($n = 2$)
- In (9.2), however, the value $n = 9$ would be necessary, a rather unrealistic proposition at the present time
- Because of this inability to reliably capture large-span behavior, n-gram performance has essentially reached a plateau

Scope Locality

- This observation has sparked interest in a variety of countermeasure, involving for instance *information aggregation* or *span extension*.
- Information aggregation increases the reliability of a word prediction by taking advantage of exemplars of other words that behave “like” this word in the particular context considered
- The trade-off, typically, is higher robustness at the expense of a loss in resolution

Scope Locality

- Span extension, which extends and/or complements the *n*-gram paradigm with information extracted from large-span units (i.e., comprising a large number of words).
- The trade-off here is in the choice of units considered for the analysis of long distance dependencies. These units tend to be either syntactic or semantic in nature

Syntactically-Driven Span Extension

- Assuming a suitable parser is available for the domain considered, syntactic information can be used to incorporate large-span constraints into the recognition
- Most recently, syntactic information has been used specifically to determine equivalence classes on the n-gram history, resulting in so-called dependency or structured LMs

Syntactically-Driven Span Extension

- In that framework, each unit is the headword of the phrase spanned by associated parse sub-tree
- The standard *n*-gram LM is then modified to operate given the last (*n*-1) *headwords* as opposed to the last (*n*-1) *words*
- As a result, the structure of the model is no longer pre-determined: which words serve as predictors depends on the dependency graph, which is a hidden variable

Semantically-Driven Span Extension

- High level semantic information can also be used to incorporate large-span constraints into the recognition
- Since by nature such information is diffused across the entire text being created, this requires the definition of a *document* as a semantically homogeneous set of sentences.
- Then each document can be characterized by drawing from a (possibly large) set of topics, usually pre-defined from a hand-labelled hierarchy, which covers the relevant semantic domain.
- The main uncertainty in this approach is the granularity required in the topic clustering procedure

Semantically-Driven Span Extension

- An alternative solution is to use long distance dependencies between word pairs which show significant correlation in the training corpus
- In the above example, suppose that the training data reveals a significant correlation between "*stocks*" and "*fell*"
- Then the presence of "*stocks*" in the document could automatically trigger "*fell*"
- Because word proximity is now irrelevant, the two phrases would lead to the same result

Semantically-Driven Span Extension

- In this approach, the pair (*stocks*, *fell*) is said to form a word trigger pair
- In practice, word pairs with high mutual information are searched for inside a windows of fixed duration
- Unfortunately, trigger pair selection is a complex issue: different pairs display markedly different behavior, which limits the potential of low frequency word triggers

Semantically-Driven Span Extension

- Recent work has sought to extend the word trigger concept by using a more comprehensive framework to handle the trigger pair selection. This is based on a paradigm originally formulated in the context of information retrieval, called *latent semantic analysis* (LSA)
- In this paradigm, co-occurrence analysis still take place across the span of an entire document, but every combination of words from the vocabulary is viewed as a potential trigger combination

Latent Semantic Analysis

- Let V, $|V| = M$, be some underlying vocabulary and T a training text corpus, comprising N articles (documents) relevant to some domain of interest
- The LSA paradigm defines a mapping between the discrete sets V, T and a continuous vector space S, whereby each word w_i in V is represented by a vector $\overline{u_i}$ in S, and each document d_j in T is represented by a vector $\overline{v_j}$ is S

Feature Extraction

- The starting point is the construction of a matrix (W) of co-occurrences between words and documents
- In marked contrast with n-gram modeling, word order is ignored, which is of course in line with the semantic nature of the approach
- This makes it an instance of the so-called “bag-of-words” paradigm, which disregards collocational information in word strings: the context for each word essentially becomes the entire document in which it appears

Feature Extraction

- This tends to involve some appropriate function of the word count in each document. Various implementations have been investigated by the information retrieval community
- Evidence point to the desirability of normalizing for document length and word entropy. Thus, a suitable expression for the (i, j) cell of W is:

$$w_{i,j} = (1 - \varepsilon_i)\frac{c_{i,j}}{n_j} \qquad (9.3)$$

where $c_{i,j}$ is the number of times w_i occurs in d_j, n_j is the total number of words present in d_j, and ε_i is the normalized entropy of w_i in the corpus T

Feature Extraction

- The global weighting implied by $1- \varepsilon_i$ reflects the fact that two words appearing with tie same count in d_j do not necessarily covey the same amount of information about the document
- If we denote by $t_i = \sum_j c_{i,j}$ the total number of times w_i occurs in T, the expression for ε_i is easily seen to be:

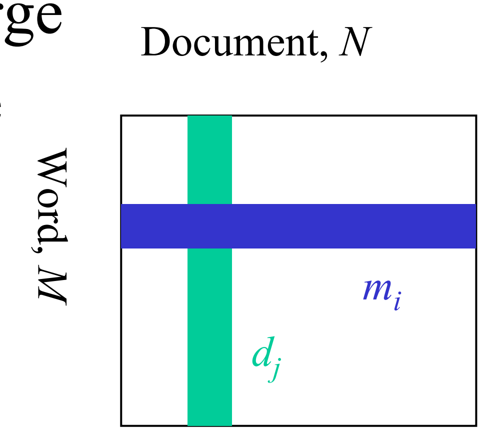
$$\varepsilon_i = -\frac{1}{\log N} \sum_{j=1}^{N} \frac{c_{i,j}}{t_i} \log \frac{c_{i,j}}{t_i} \qquad (9.4)$$

Feature Extraction

- By definition, $0 \leqq \varepsilon_i \leqq 1$, with equality if and only if $c_{i,j} = t_i$ and $c_{i,j} = t_i/N$, respectively
- A value of ε_i close to 1 indicates a word distributed across many documents throughout the corpus, while a value of ε_i close to 0 means that the word is present only in a few specific documents
- The global weight $1- \varepsilon_i$ is therefore a measure of the indexing power of the word w_i

Singular Value Decomposition

- The (M x N) word-document matrix W defines two vector representations for the words and the documents. Each word w_i can be uniquely associated with a row vector of dimension N, and each document d_j can be uniquely associated with a column vector of dimension M
- Unfortunately, this is unpractical for three reasons
  - The dimensions M and N can be extremely large
  - The vectors w_i and d_j are typically very sparse
  - The two spaces are distinct from on another



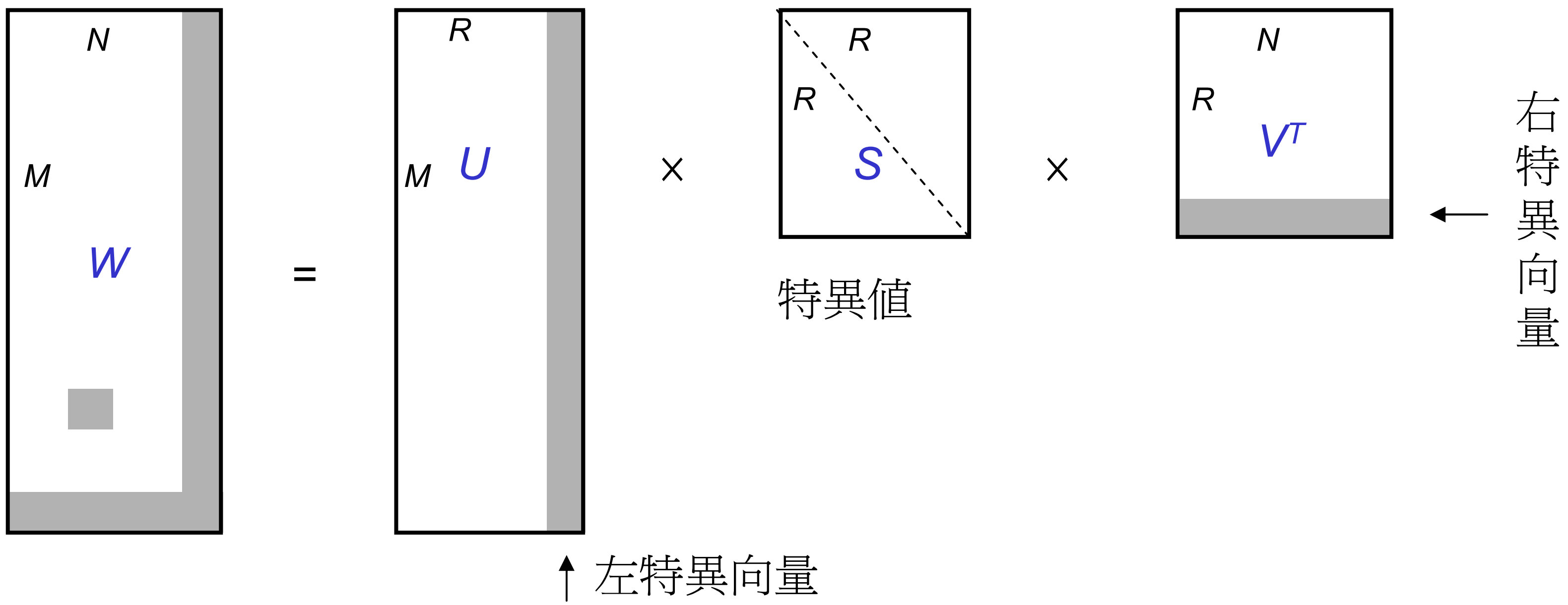
Singular Value Decomposition

- To address these issues, one solution is to perform the (order-R) singular value decomposition (SVD) of W:

$$W \approx \hat{W} = USV^T \tag{9.5}$$

where U is the (M x R) left singular matrix with row vectors u_i ($1 \leqq i \leqq M$), S is the (R x R) diagonal matrix of singular value $s_1 \geqq s_2 \geqq \ldots \geqq s_R > 0$, V is the (N x R) right singular matrix with row vectors v_j ($1 \leqq j \leqq N$), $R << \min(M,N)$ is the order of the decomposition

Singular Value Decomposition



LSA Feature Space

- In the continuous vector space S, each word $w_i \in V$ is represented by the associated word vector of dimension R, $\overline{u_i} = u_i S$, and each document $d_j \in T$ is represented by the associated document vector of dimension R, $\overline{v_j} = v_j S$
- Since the matrix W embodies all structural associations between words and documents for a given training corpus, WW^T characterizes all co-occurrences between words, and W^TW characterizes all co-occurrences between documents

Word Clustering

- Expanding WW^T using the SVD expression (9.5), we obtain:

$$WW^T = USV^T \times VS^T U^T = US^2 U^T \qquad (9.6)$$

- Since S is diagonal, a natural metric to consider for the "closeness" between words is therefore the cosine of the angle between $u_i S$ and $u_j S$:

$$K(w_i, w_j) = \cos(\bar{u}_i, \bar{u}_j) = \frac{u_i S^2 u_j^T}{\|u_i S\| \|u_j S\|} \qquad (9.7)$$

for any $1 \leqq i, j \leqq M$

Word Clustering

- A value of $K(w_i, w_j) = 1$ means the two words always occur in the same semantic context, while a value of $K(w_i, w_j) \leqq 1$ means the two words are used in increasingly different semantic contexts
- While (9.7) does not define a bona fide distance measure in the space S, it easy leads to one. For example, over the interval $[0, \pi]$, the measure:

$$D(w_i, w_j) = \cos^{-1} K(w_{i,} w_j) \qquad (9.8)$$

Word Cluster Example

- A corpus of $N = 21{,}000$ documents, vocabulary of $M = 23{,}000$ words, and the word vectors in the resulting LSA space were clustered into 500 disjoint clusters using a combination of **K**-means and bottom-up clustering
- Figure 9.2 shows two clusters
- Polysemy (some words seem to be missing)
  - drawing from cluster 1, (drawing a conclusion)
  - rule from cluster 2, (breaking a rule)
- "hysteria" from cluster 1 and "here" from cluster 2 are the unavoidable outliers at the periphery of the clusters

Cluster 1
Andy, antique, antiques, art, artist, artist's, artists, artworks, auctioneers, Christie's, collector, drawings, gallery, Gogh, fetched, hysteria, masterpiece, museums, painter, painting, paintings, Picasso, Pollock, reproduction, Sotheby's, van, Vincent, Warhol

Cluster 2
appeal, appeals, attorney, attorney's, counts, court, court's, courts, condemned, convictions, criminal, decision, defend, defendant, dismisses, dismissed, hearing, here, indicted, indictment, indictments, judge, judicial, judiciary, jury, juries, lawsuit, leniency, overturned, plaintiffs, prosecute, prosecution, prosecutions, prosecutors, ruled, ruling, sentenced, sentencing, suing, suit, suits, witness

FIGURE 9.2
Word Cluster Example (After [2]).

Document Clustering

- Proceeding in a similar fashion at the document level, we obtain:

$$W^T W = VS^T U^T \times USV^T = VS^2 V^T \qquad (9.9)$$

- For $1 \leqq i, j \leqq N$, leads to the same functional form as (9.7)

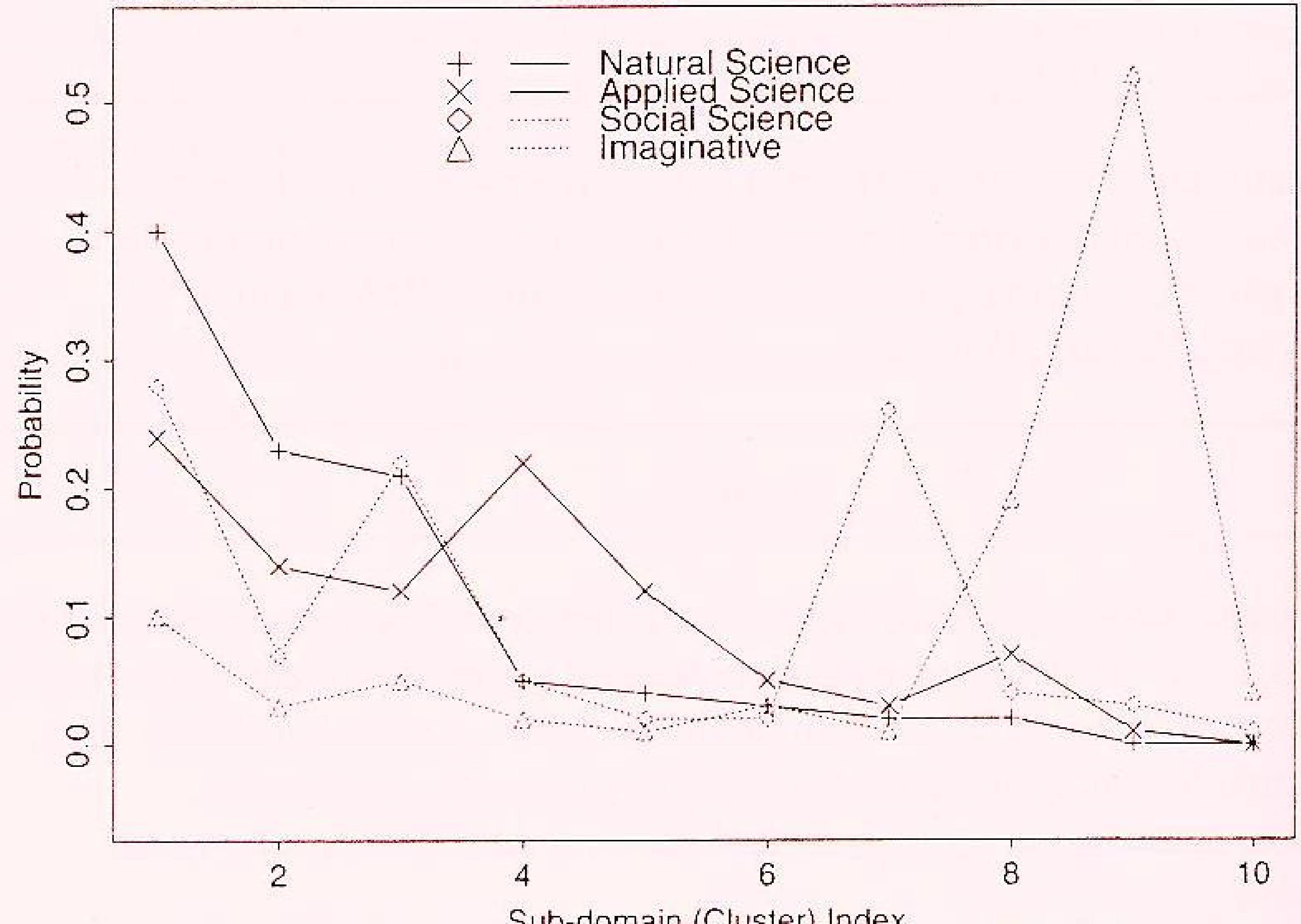
$$K(d_i, d_j) = \cos(\bar{v}_i, \bar{v}_j) = \frac{v_i S^2 v_j^T}{\|v_i S\| \|v_j S\|} \qquad (9.10)$$

Document Cluster Example

- This experiment was conducted on the British National Corpus, a heterogeneous corpus which contains a variety of hand-labelled topics
- The LSA framework was used to partition BNC into distinct clusters, and the sub-domains so obtained were compared with the hand-labelled topics provided with the corpus
- This comparison war conducted in an objective manner by evaluating two different mixture trigram LMs: one built from the LSA sub-domain, and the other from the hand-labelled topics

Document Cluster Example

- As the perplexities obtained were very similar, it showed that the automatic partitioning performed using LSA was indeed semantically coherent
- Figure 9.3 plots the distributions of 4 of the hand-labelled BNC topics against the 10 document sub-domains automatically derived using LSA. Although it is clear that the data-driven sub-domains do not exactly match the hand-labeling, LSA document clustering in this example still seems reasonable
  - The distribution of natural science topic is relatively close to the distribution of applied science topic, but quit different from the two other topic distributions
  - From this standpoint, the data-driven LSA cluster appear to adequately cover the semantic space



Probability Distributions of Four BNC Topics Against LSA Document Clusters

Semantic Classification

- Semantic classification determines, for a given document, which one of several predefined topics, the document is most closely aligned with
  - Such document will not (normally) have been seen in the training corpus
  - We need to extend the LSA framework accordingly

Framework Extension

Let us denote the new document by $\widetilde{d}_p$, where the tilde symbos (~) reflects the fact that $p > N$.

This vector $\widetilde{d}_p$, as a column vector of dimension M, can be thought of as an additional column of the matrix W

Provided the matrices U and S do not change, the SVD expansion (9.5) implies:

$$\widetilde{d}_p = US\widetilde{v}_p^T \qquad (9.11)$$

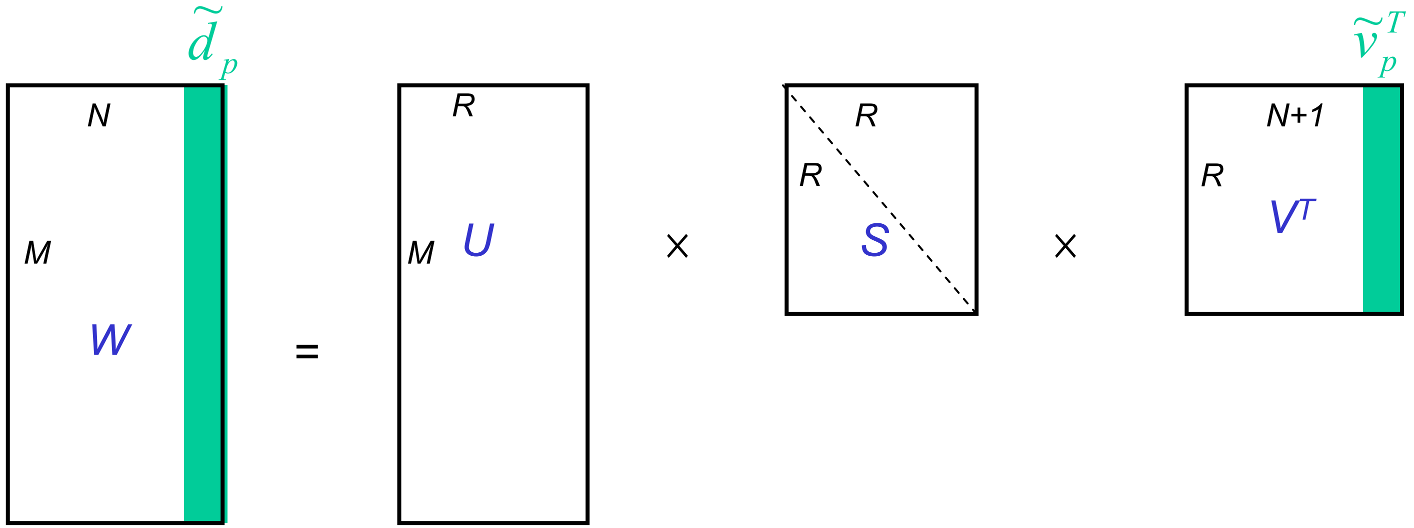
where the R-dimensional vector $\widetilde{v}_p^T$ acts as an additional column of the matrix V^T

Framework Extension

- This in turn leads to the definition:

$$\tilde{\tilde{v}}_p = \tilde{v}_p S = \tilde{d}_p^T U \qquad (9.12)$$

$\tilde{\tilde{v}}_p$ is referred to as a *pseudo document vector*



Semantic Inference

- Suppose that each document cluster D_l can be uniquely associated with a particular action in the task. Then the centroid of each cluster can be viewed as the *semantic anchor* of this action in the LSA space
- An unknown word sequence (treated as a new "document") can thus be mapped onto an action by evaluating the distance between that "document" and each semantic anchor.
- We refer to this approach as *semantic inference*

Semantic Inference

- Consider an application with N=4 actions (documents), each associated with a unique command:
  - (i) “what is the time”
  - (ii) “what is the day”
  - (iii) “what time is the meeting”
  - (iv) “cancel the meeting”
- This simple example, with a vocabulary of only M=7 words, is designed such that “*what*” and “*is*” always co-occur, “*the*” appears in all four commands, only (ii) and (iv) contain a unique word, and (i) is a proper subset of (iii)
- (7*4) word-document matrix, perform SVD

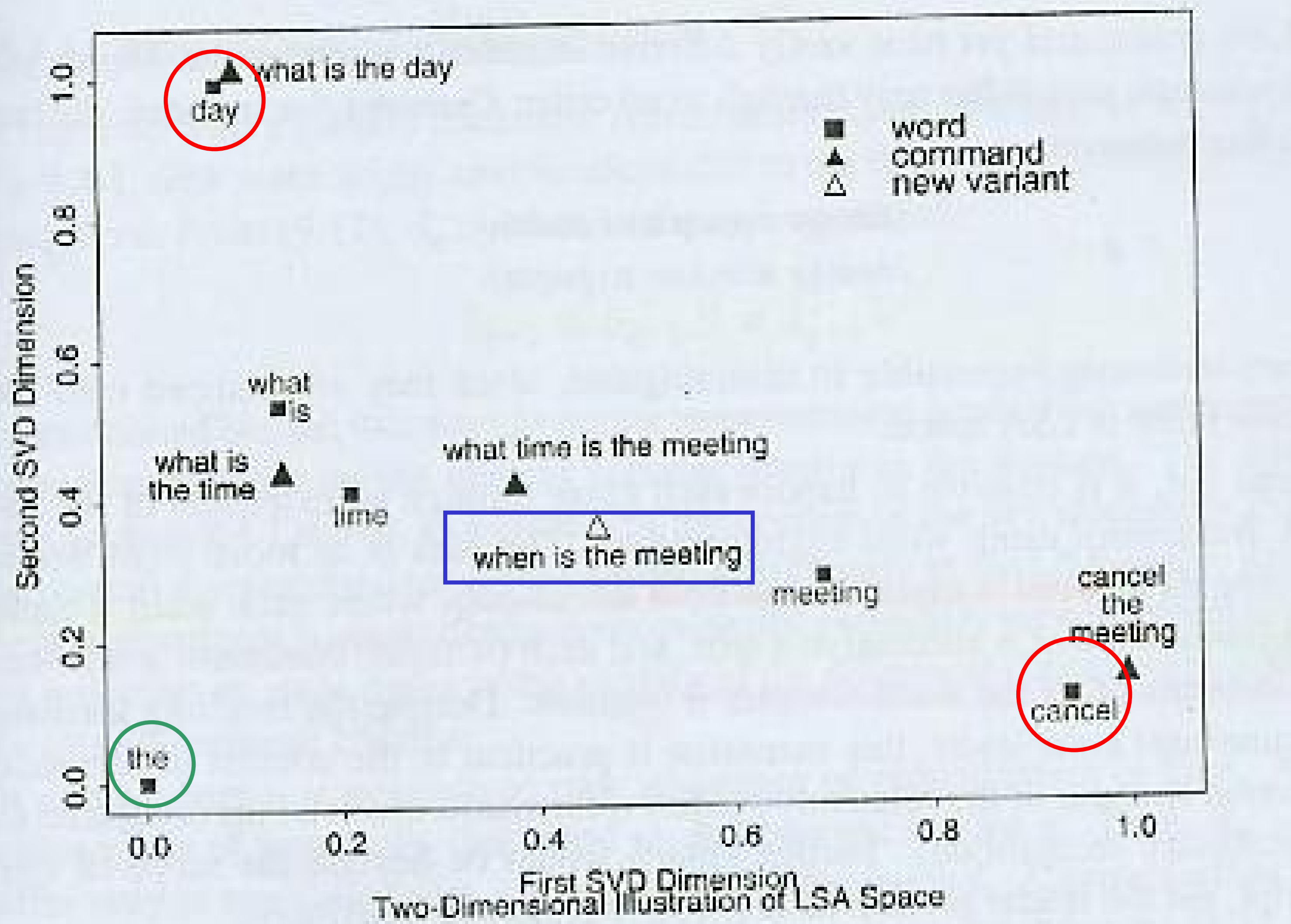


FIGURE 9.4

An Example of Semantic Inference for Command and Control ($R = 2$).

Caveats

- LSA pays no attention to the order of words in sentences, which makes it ideally suited to capture large-span semantic relationships
- By the same token, however, it is inherently unable to capitalize on the local (syntactic, pragmatic) constrains present in the language

change popup to window
change window to popup

- Which are obviously impossible to disambiguate, since they are mapped onto the *exact same point* in LSA space

Caveats

- As it turns out, it is possible to handle such cases through an extension of the basic LSA framework using word agglomeration.
  - Words → word *n*-tuples
    (agglomeration of *n* successive words)
  - Documents → *n*-tuple documents
    (each *n*-tuple document is expressed in terms of all the word *n*-tuples it contains)

N-gram + LSA Language Modeling
LSA Component

- Let w_q denote the word about to be predicted, and $H_{q\text{-}1}$ the admissible LSA history (context) for this particular word.

  This notation translates a causality restriction of the context to $\tilde{d}_{q-1}$, the current document so far (i.e., up to word w_{q-1})

  Thus, in general terms, the LSA LM probability is given by :

$$\Pr(w_q \mid H_{q-1}, S) = \Pr(w_q \mid \tilde{d}_{q-1}) \qquad (9.14)$$

Pseudo document representation

From (9.12), $\tilde{d}_{q-1}$ leads to the representation :

$$\tilde{\tilde{v}}_{q-1} = \tilde{v}_{q-1} S = \tilde{d}_{q-1}^{T} U \qquad (9.15)$$

- As q increases, the content of the new document grows and the pseudo document vector moves around accordingly in the LSA space
- Assuming the new document is semantically homogeneous, eventually we can expect the resulting trajectory to settle down in the vicinity of the document cluster corresponding to the closest semantic content

Pseudo document representation

$$\tilde{d}_q = \frac{n_q - 1}{n_q} \tilde{d}_{q-1} + \frac{1 - \varepsilon_i}{n_q} [0 \ldots 1 \ldots 0]^T \qquad (9.16)$$

- Where the "1" appears at coordinate *i*. This is turn implies, from (9.15):

$$\tilde{\tilde{v}}_q = \tilde{v}_q S = d_{q-1}^T U = \frac{1}{n_q} \left[(n_q - 1) \tilde{\tilde{v}}_{q-1} + (1 - \varepsilon_i) u_i \right] \qquad (9.17)$$

LSA Probability

- A natural metric to consider for the "closeness" between word w_i and document d_j is the cosine of the angle between $u_i S^{1/2}$ and $v_j S^{1/2}$.
  Applying the same reasoning to pseudo documents, we arrive at:

$$K(w_q, \tilde{d}_{q-1}) = \cos(u_q S^{1/2}, \tilde{v}_{q-1} S^{1/2}) = \frac{u_q S \tilde{v}_{q-1}^T}{\left\| u_q S^{1/2} \right\| \left\| \tilde{v}_{q-1} S^{1/2} \right\|} \qquad (9.18)$$

for any q indexing a word in the text data

A value of $K = 1$ means that $\tilde{d}_{q-1}$ is a strong semantic predictor of w_q, while a value of $K < 1$ means that the history carries increasingly less information about the current word

LSA Probability

Intuitively, $\Pr(w_q \mid \tilde{d}_{q-1})$ reflects the "relevance" of word w_q to the admissible history, as observed through $\tilde{d}_{q-1}$. As such, it will be highest for words whose meaning aligns most closely with the semantic favric of $\tilde{d}_{q-1}$ (i.e., relevant "content" words), and lowest for words which do not convey any particular information about this fabric (e.g., "function" works like "*the*").

- Conventional *n*-gram
  - Assign higher probabilities to (frequent) function words than to (rarer) content words
- Hence, the attractive synergy potential between the two paradigms

Integration with N-grams

$$\Pr(w_q \mid H_{q-1}^{(n+l)}) = \Pr(w_q \mid H_{q-1}^{(n)}, H_{q-1}^{(l)}) \qquad (9.19)$$

where H_{q-1} denotes some suitable history for word w_q, and the superscripts $^{(n)}, ^{(l)}$, and $^{(n+l)}$ refer to the n-gram component ($w_{q-1}w_{q-2}...w_{q-n+1}$, with $n > 1$), the LSA component ($\tilde{d}_{q-1}$), and the integration thereof, respectively. This expression can be rewritten as:

$$\Pr(w_q \mid H_{q-1}^{(n+l)}) = \frac{\Pr(w_q, H_{q-1}^{(l)} \mid H_{q-1}^{(n)})}{\sum_{w_i \in V} \Pr(w_i, H_{q-1}^{(l)} \mid H_{q-1}^{(n)})} \qquad (9.20)$$

Integration with N-grams

- Expanding and re-arranging, the numerator of (9.20) is seen to be:

$$\Pr(w_q, H_{q-1}^{(l)} \mid H_{q-1}^{(n)}) =$$

$$\Pr(w_q \mid H_{q-1}^{(n)}) \cdot \Pr(H_{q-1}^{(l)} \mid w_q, H_{q-1}^{(n)}) = \tag{9.21}$$

$$\Pr(\boxed{w_q} \mid \boxed{w_{q-1} w_{q-2} \cdots w_{q-n+1}}) \cdot \Pr(\boxed{\tilde{d}_{q-1}} \mid w_q \boxed{w_{q-1} w_{q-2} \cdots w_{q-n+1}})$$

Now we make the assumption that the probability of the document history given the current word is not affected by the immediate context preceding it

For a given word, different syntactic constructs (immediate context) can be used to carry the same meaning (document history)

Integration with N-grams

- As a result, the integrated probability becomes:

$$\Pr(w_q \mid H_{q-1}^{(n+l)}) =$$

$$\frac{\Pr(w_q \mid w_{q-1} w_{q-2} \cdots w_{q-n+1}) \cdot \Pr(\tilde{d}_{q-1} \mid w_q)}{\sum_{w_i \in V} \Pr(w_i \mid w_{q-1} w_{q-2} \cdots w_{q-n+1}) \cdot \Pr(\tilde{d}_{q-1} \mid w_i)} \qquad (9.22)$$

Integration with N-grams

The dependence of (9.22) on the LSA probability calculated earlier can be expressed explicitly by using Bayes' rule to get $\Pr(\tilde{d}_{q-1} \mid w_q)$ in terms of $\Pr(w_q \mid \tilde{d}_{q-1})$.

$$\Pr(w_q \mid H_{q-1}^{(n+l)}) =$$

$$\frac{\Pr(w_q \mid w_{q-1} w_{q-2} \cdots w_{q-n+1}) \cdot \dfrac{\Pr(w_q \mid \tilde{d}_{q-1})}{\Pr(w_q)}}{\sum_{w_i \in V} \Pr(w_i \mid w_{q-1} w_{q-2} \cdots w_{q-n+1}) \cdot \dfrac{\Pr(w_i \mid \tilde{d}_{q-1})}{\Pr(w_i)}} \qquad (9.23)$$

$n > 1$. If n=1, (9.23) degenerates to (9.14)

Context Scope Selection

- During training, the context scope is fixed to be the current document.
- During recognition, the concept of "current document" is ill-defined, because
  - (i) its length grows with each new word
  - (ii) it is not necessarily clear at which point completion occurs
- As a result, a decision has to be made regarding what to consider "current," versus what to consider part of an earlier (presumably less relevant) document

Context Scope Selection

- A straightforward solution is to limit the size of the history considered, so as to avoid relying on ole, possibly obsolete fragments, to construct the current context
- Alternatively, it is possible to assume an exponential decay in the relevance of the context
  - In this solution, exponential forgetting is used to progressively discount older utterances

$$\tilde{\bar{v}}_q = \frac{1}{n_q}\left[\lambda\left(n_q - 1\right)\tilde{\bar{v}}_{q-1} + \left(1 - \varepsilon_i\right)u_i\right] \qquad (9.24)$$

$0 < \lambda \leq 1$. λ is chosen according to the expected heterogeneity of the session

Word Smoothing

- Using the set of word clusters C_k, $1 \leqq k \leqq K$, leads to word-based smoothing. Expand (9.14) as follows:

$$\Pr\left(w_q \mid \tilde{d}_{q-1}\right) = \sum_{k=1}^{K} \Pr\left(w_q \mid C_k\right) \Pr\left(C_k \mid \tilde{d}_{q-1}\right) \qquad (9.25)$$

$\Pr\left(C_k \mid \tilde{d}_{q-1}\right)$ is qualitatively similar to (9.14) and can therefore be obtained with the help of (9.18), by simply replacing the representation of the word w_q by that of the centroid of word cluster C_k

$\Pr\left(w_q \mid C_k\right)$ denotes on the "closeness" of w_q relative to this (word) centroid.

Word Smoothing

- The behavior of the model (9.25) depends on the number of word clusters defined in the space S
- Two special cases arise at the extremes of the cluster range
  - As many classes as words in the vocabulary (K=M), then with the convention that $P(w_i|C_j)=\delta_{ij}$, (9.25) simply reduces to (9.14)
  - All the words are in a single class (K=1), the model become maximally smooth: the influence of specific semantic events disappears, leaving only a residual vocabulary effect to take into account

Document Smoothing

- Exploiting instead the set of document clusters D_l, $1 \leqq l \leqq L$, leads to document-based smoothing. The expansion is similar:

$$\Pr\left(w_q \mid \tilde{d}_{q-1}\right) = \sum_{k=1}^{K} \Pr\left(w_q \mid D_l\right) \Pr\left(D_l \mid \tilde{d}_{q-1}\right) \qquad (9.26)$$

$\Pr\left(w_q \mid D_l\right)$ is qualitatively similar to (9.14) and can therefore be obtained with the help of (9.18).

$\Pr\left(D_l \mid \tilde{d}_{q-1}\right)$, it depends on the "closeness" of $\tilde{d}_{q-1}$ relative to the centroid of document cluster D_l

Joint Smoothing

$$\Pr(w_q \mid \widetilde{d}_{q-1}) = \sum_{k=1}^{K}\sum_{l=1}^{L} \Pr(w_q \mid C_k, D_l)\Pr(C_k, D_l \mid \widetilde{d}_{q-1}) \qquad (9.28)$$

Which, for tractability, can be approximated as:

$$\Pr(w_q \mid \widetilde{d}_{q-1}) = \sum_{k=1}^{K}\sum_{l=1}^{L} \Pr(w_q \mid C_k)\Pr(C_k \mid D_l)\Pr(D_l \mid \widetilde{d}_{q-1}) \qquad (9.29)$$

Some summarize

- Any of the expressions (9.14), (9.25), (9.26), or (9.29) can be used to compute (9.23)

$$\Pr(w_q \mid H_{q-1}, S) = \Pr(w_q \mid \widetilde{d}_{q-1}) \qquad (9.14)$$

$$\Pr\left(w_q \mid \widetilde{d}_{q-1}\right) = \sum_{k=1}^{K} \Pr\left(w_q \mid C_k\right) \Pr\left(C_k \mid \widetilde{d}_{q-1}\right) \qquad (9.25)$$

$$\Pr\left(w_q \mid \widetilde{d}_{q-1}\right) = \sum_{k=1}^{K} \Pr\left(w_q \mid D_l\right) \Pr\left(D_l \mid \widetilde{d}_{q-1}\right) \qquad (9.26)$$

$$\Pr(w_q \mid \widetilde{d}_{q-1}) = \sum_{k=1}^{K} \sum_{l=1}^{L} \Pr(w_q \mid C_k) \Pr(C_k \mid D_l) \Pr(D_l \mid \widetilde{d}_{q-1}) \qquad (9.29)$$

Experiments
Experimental Conditions

- T, N = 87,000 documents spanning the years 1987 to 1989, 42M words
- V, M = 23,000 words
- Test set, 496 sentences uttered by 12 native speakers of English
- Acoustic training was performed using 7,200 sentences of data uttered by 84 speakers (WSJ0 SI-84)
- Baseline: Bigram 16.7%, Trigram 11.8%
- R = 125, K = 100 word clusters, L = 1 document cluster

Experimental Results

TABLE 9.1

Word Error Rate (WER) Results Using Hybrid Bi-LSA and Tri-LSA Models.

Word Error Rate <WER Reduction>	Bigram $n = 2$	Trigram $n = 3$
Conventional n-Gram	16.7 %	11.8 %
Hybrid, No Smoothing	14.4 % <14 %>	10.7 % < 9 %>
Hybrid, Document Smoothing	13.4 % <20 %>	10.4 % <12 %>
Hybrid, Word Smoothing	12.9 % <23 %>	9.9 % <16 %>
Hybrid, Joint Smoothing	13.0 % <22 %>	9.9 % <16 %>

- Such results show that the hybrid n-gram+LSA approach is a promising avenue for incorporating large-span semantic information into n-gram modeling

Context Scope Selection

- By design, the test corpus is constructed with no more than three or four consecutive sentences extracted from a single article. Overall, it comprises 140 distinct document fragments, which means that each speaker speaks, on average, about 12 different “mini-documents.” As a result, the context effectively changes every 60 words or so.
- $\lambda = 1$ to $\lambda = 0.95$, in decrements of 0.01

TABLE 9.2
Influence of Context Scope Selection on Word Error Rate.

Word Error Rate <WER Reduction>	Bi-LSA with Word Smoothing
$\lambda = 1.0$	14.5 % <13 %>
$\lambda = 0.99$	13.6 % <18 %>
$\lambda = 0.98$	13.2 % <21 %>
$\lambda = 0.975$	12.9 % <23 %>
$\lambda = 0.97$	13.0 % <22 %>
$\lambda = 0.96$	13.1 % <22 %>
$\lambda = 0.95$	13.5 % <19 %>

Cross-Domain Training

- In the previous section, both LSA and *n*-gram components of the hybrid LM were trained on exactly the same data
  - How critical the selection of the LSA training data is to the performance of the recognizer
- Unsmoothed model (9.14), the same underlying vocabulary V, bigram, and repeated the LSA training on non-WSJ (Associated Press (AP))data from the same general period
  - (i) T_1, N_1 = 84,000 documents from 1989, 44M words
  - (ii) T_2, N_2 = 155,000 documents from 1988-89, 80M words
  - (iii) T_3, N_3 = 224,000 documents from 1988-90, 117M words

Cross-Domain Training

TABLE 9.3
Model Sensitivity to LSA Training Data.

Word Error Rate <WER Reduction>	Bi-LSA with No Smoothing
T_1: $N_1 = 84,000$	16.3 % <2 %>
T_2: $N_2 = 155,000$	16.1 % <3 %>
T_3: $N_3 = 224,000$	16.0 % <4 %>

- First, the performance improvement in all case is much smaller than the 14% reduction observed in Table 9.1, on the average, the hybrid model trained on AP data is about four times less effective than that trained on WSJ data.
- This suggests a relatively high LSA sensitivity to the domain considered

Cross-Domain Training

- Second, the overall performance does not improve appreciably with more training data
- This supports the conjecture that LSA is sensitive not just to the general training domain, but also to the particular style of composition.
- On the positive side, this bodes well for rapid adaptation to cross-domain data, provided a suitable adaptation framework can be derived.

Discussion

- LSA is inherently more adept at handling content words than function words.
- As is well-known, a substantial proportion of speech recognition errors come from function words, because of their tendency to be shorter, not well articulated, an acoustically confusable
- Even within a well-specified domain, syntactically-driven span extension techniques may be a necessary complement to the hybrid approach
  - Headword-based *n*-gram

Conclusion

- Statistical *n*-grams are by nature limited to the capture of linguistic phenomena spanning at most *n* words
- Semantically-driven span extension framework based on the LSA paradigm
- Hybrid *n*-gram + LSA model
- LSA shows sensitivity to both the training domain and the style of composition