Topic Modeling in Word Prediction

Keith Trnka

University of Delaware
SIGAI
May 1st, 2006

Who's on the project

- Grad students
  - Keith Trnka
  - Debra Yarrington
  - John McCaw
  - Amit Hetawal
- Everyone else
  - Kathy McCoy
  - Christopher Pennington (AgoraNet, Inc)
  - Other people

Augmentative and Alternative Communiction (AAC)

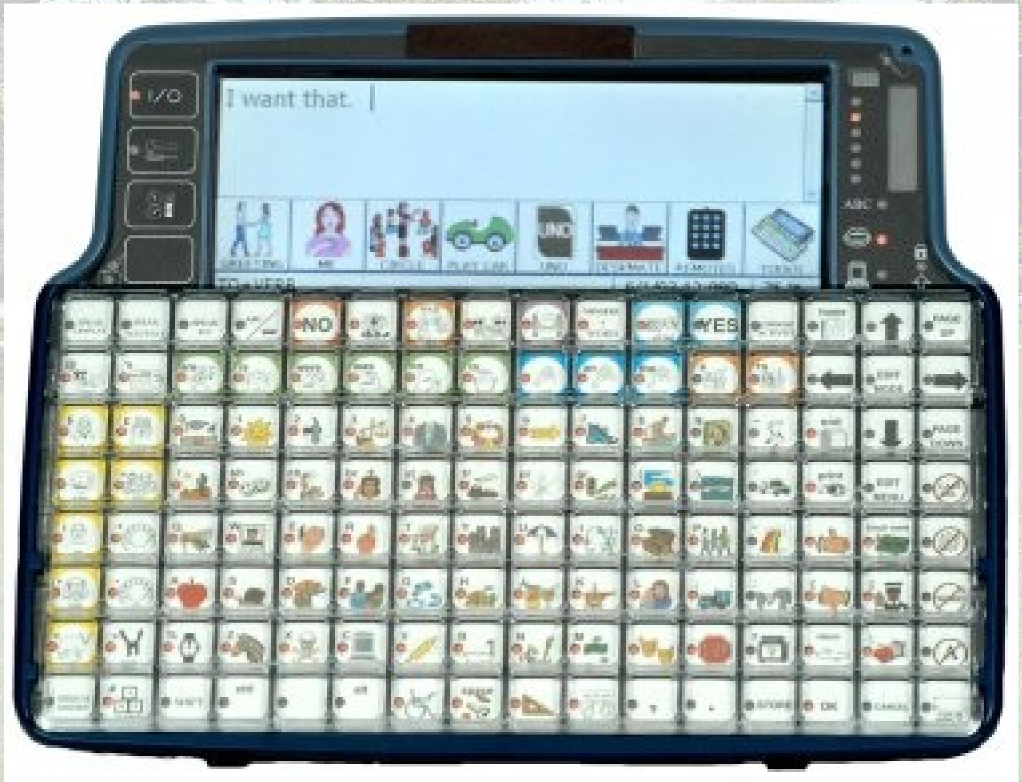
- People with communication disabilities
- Many are unable to speak
- Multiple disabilities common
  - Motor impairment
  - Cognitive impairment
- Some are illiterate

Non-electronic Solutions

- Sign language
- Word/picture boards

Electronic Solutions

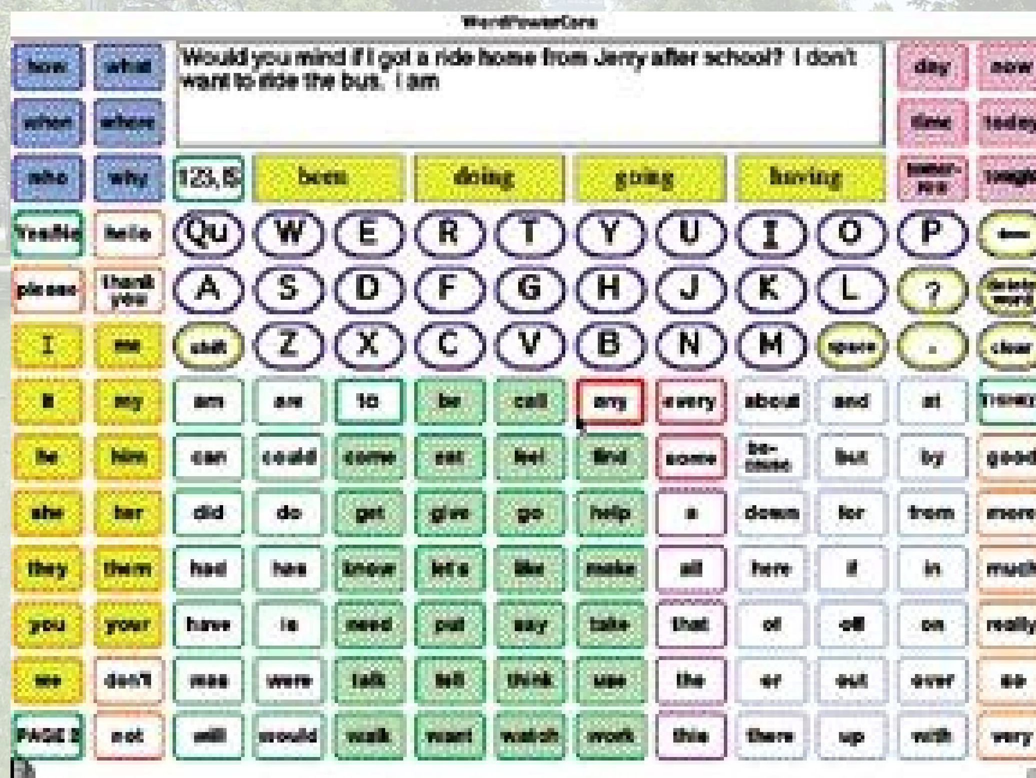
- Common:
  Text entry with speech synthesis



Prentke-Romich Company's *Pathfinder Plus*

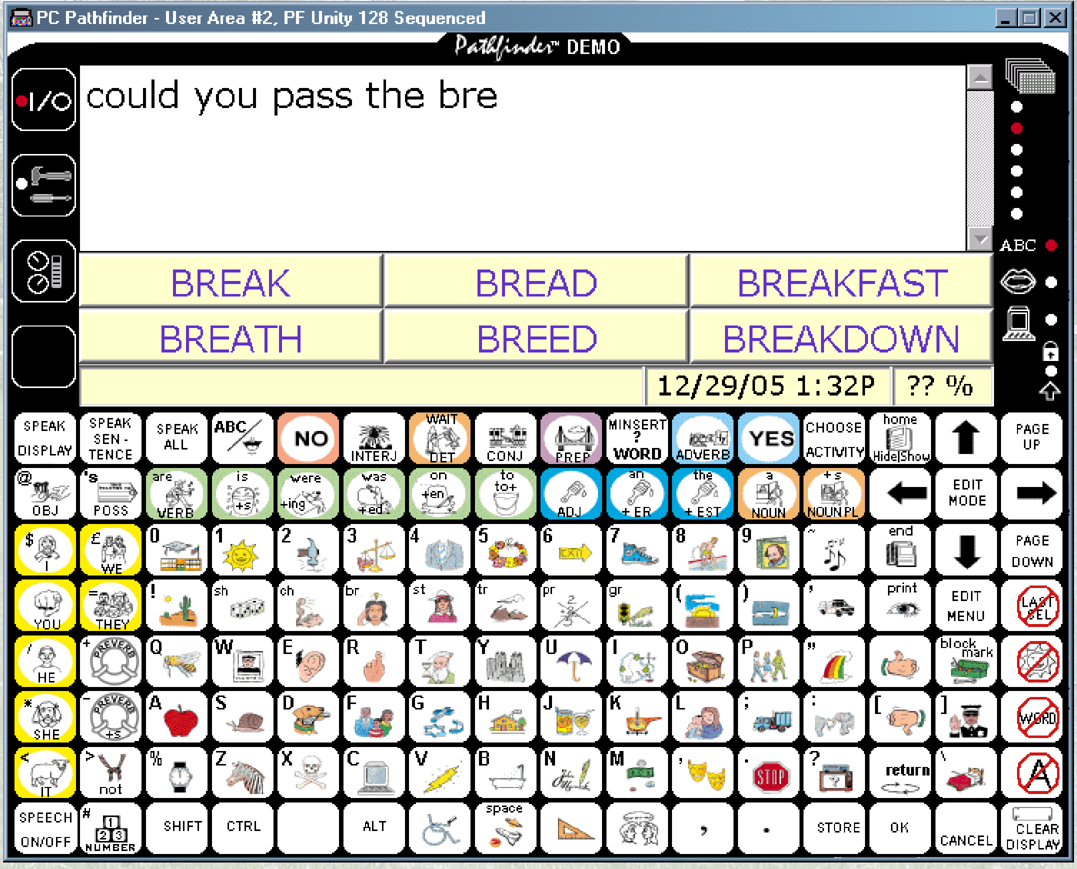
Electronic Solutions

- User input – what do they input
  - Letter selection (semi-standard keyboard)
  - Word selection



Electronic Solutions

- User input – what do they input
  - Icon selection
  - Phrase selection



Electronic Solutions

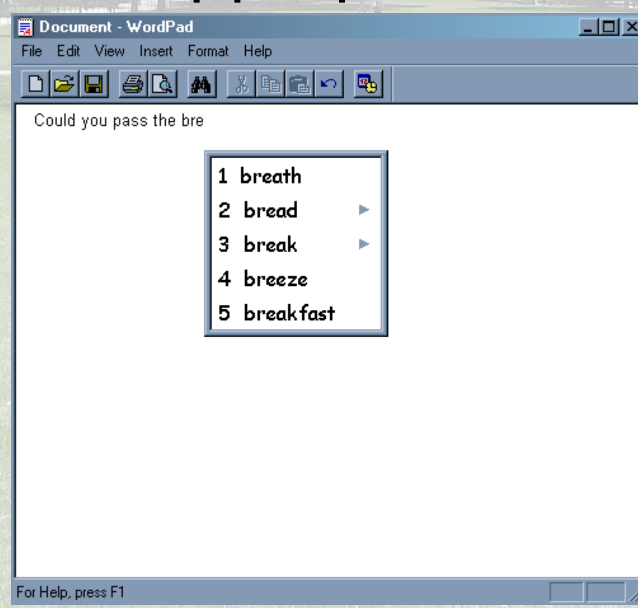
- User input – how do they input
  - Direct selection
  - Row-column scanning
  - Fingers, hands, head pointer, head motion, eye gaze

Electronic Solutions

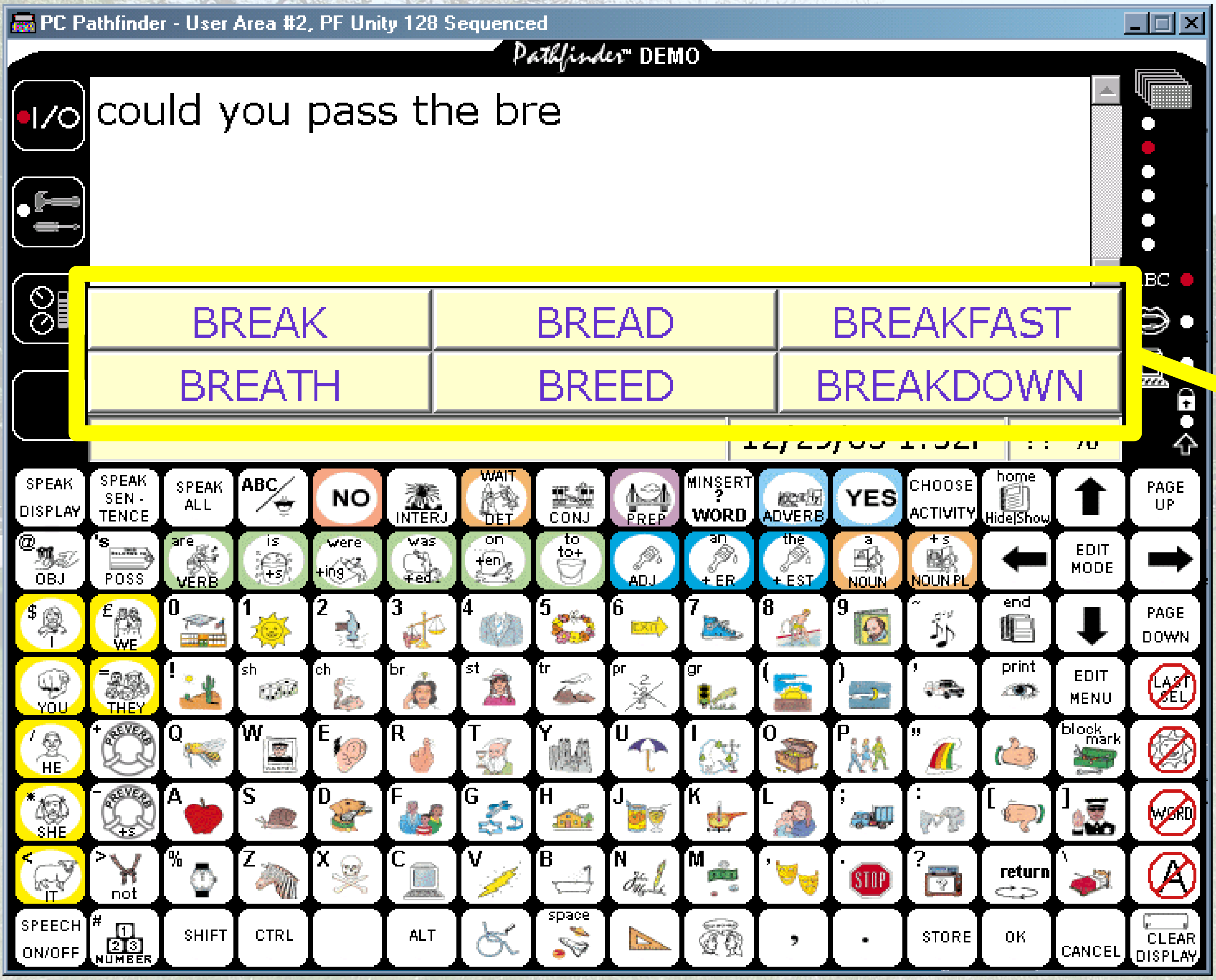
- Making it faster
  - Abbreviation expansion
    - Traditional/written: acronyms, contractions, ellipsis, etc
    - Instant Messaging: brb, afaik, iirc
    - Dynamic: dynmc, dnsr, airpl
    - Practice: abbreviation lists

Electronic Solutions

- Making it faster (cont'd)
  - Buttons/hierarchy for common phrases
  - Buttons for common words
    - Static list of common words – *core vocabulary*
    - Dynamic list of appropriate words – word prediction



Buttons for Common Words

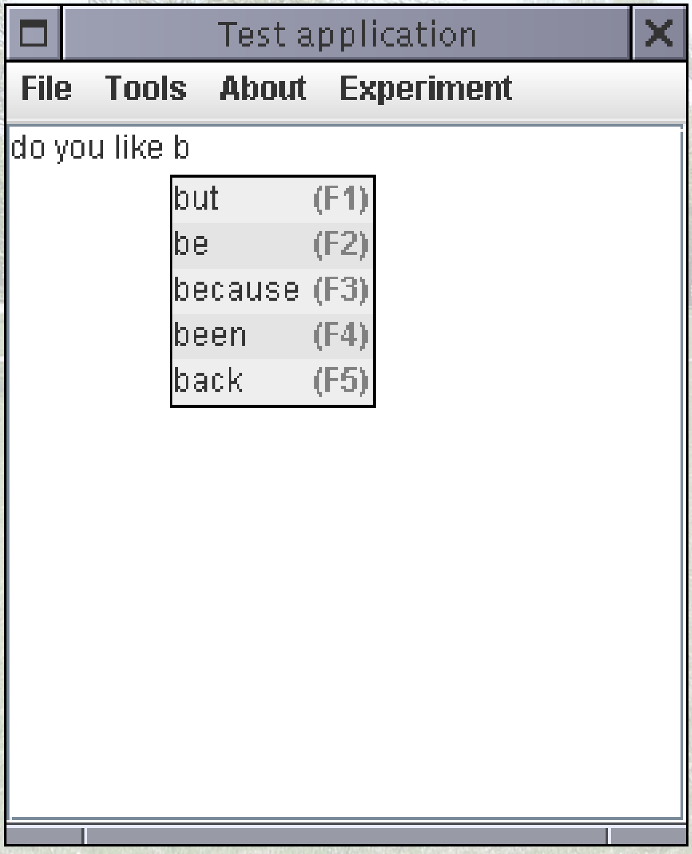


Prediction window (size 6)

Letter entry and core vocabulary via icon sequences

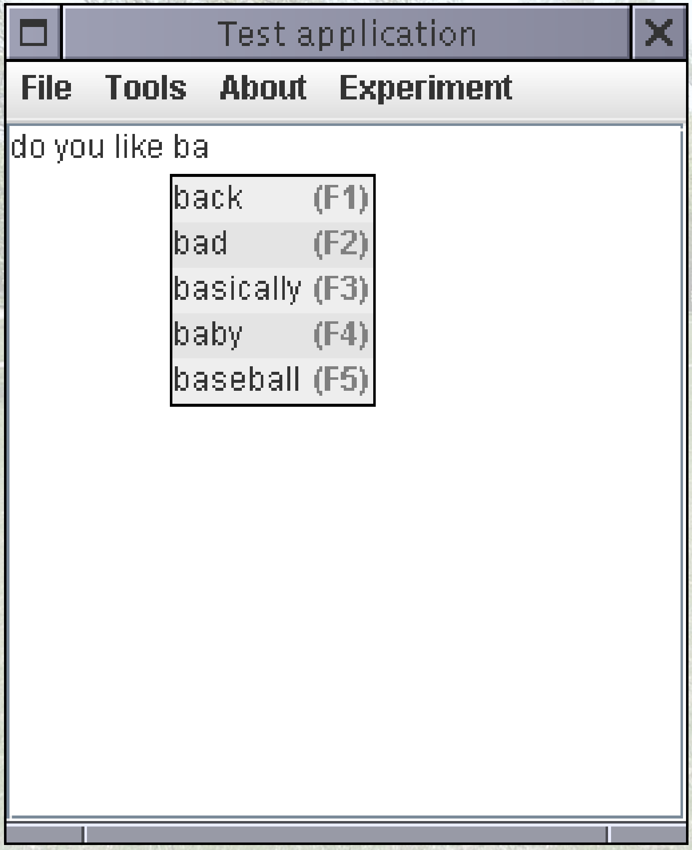
Word Prediction

- Suppose a user is asking a friend “do you like baseball games?”



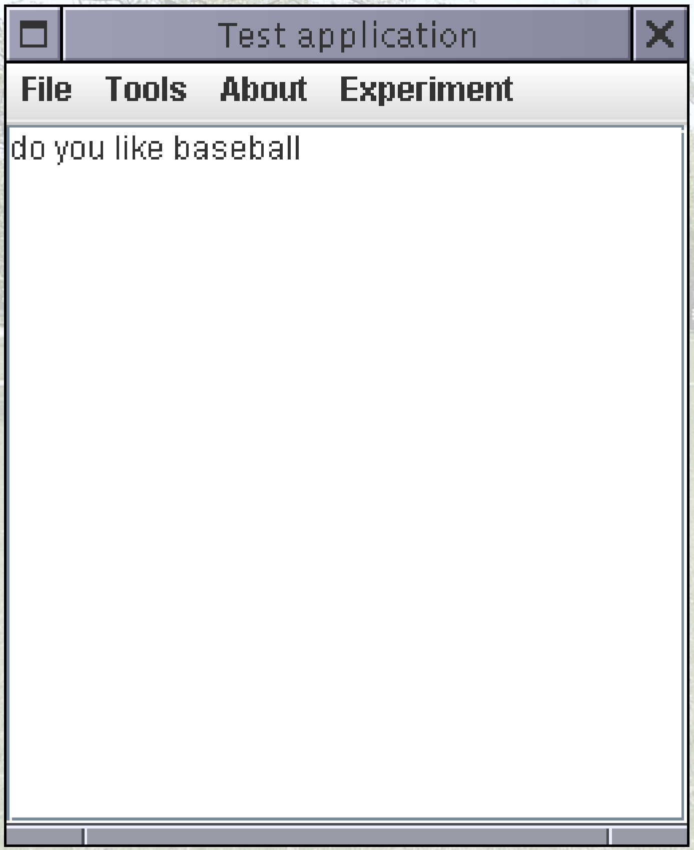
Word Prediction

- User pressed 'a'



Word Prediction

- User pressed 'F5'



Word Prediction

- Advantages
  - Low cognitive effort – very little time to learn
  - Very little screen real estate
  - Doesn't require memorization*
  - Reduces a user interface problem to a NLP problem
  - Can augment a core/fringe split system by focusing on fringe

Word Prediction

- Disadvantages
  - Requires same-domain training data to perform well
  - Requires some perceptual effort – distractions

Word Prediction

- Practical issues
  - Number of words to predict
    - 5-7 is common
  - Placement and orientation of the predictions – compounded by the rest of the GUI
    - Vertical lists are easier to glance at quickly
    - Horizontal lists can be placed between the keyboard and editing area
  - How to select predicted words
    - Physical devices are often touchscreens
  - Trusting the predictions

Word Prediction

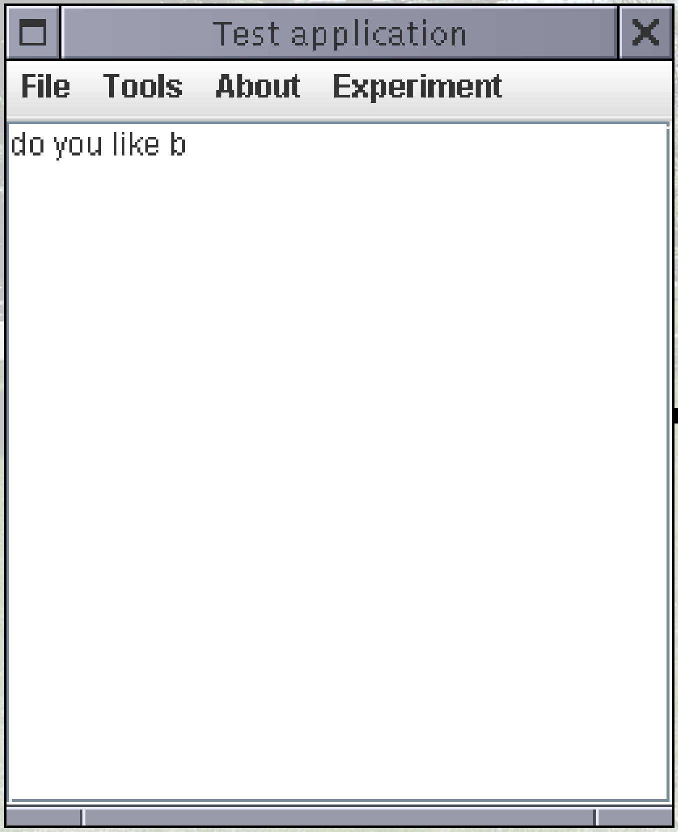
- Practical issues (cont'd)
  - Static vs. dynamic language model
    - Static – the language model doesn't adapt to the user
      - Lower cognitive demands
      - Allows for memorization
    - Dynamic – the language model adapts to what the user types
      - Doesn't always allow for memorization
      - Higher computational demands (generally)
      - Often better predictions

Word Prediction

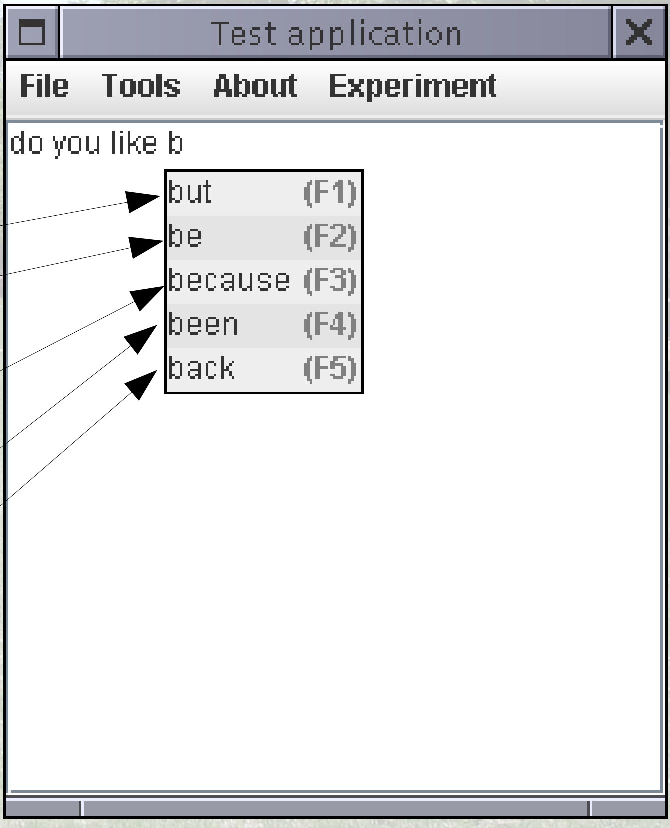
- Research issues
  - Company-researcher stances on word prediction
  - Core vs. Fringe vocabulary
    - Core vocabulary is company-specific
    - Fringe words are the rest of the vocabulary, often typed out using word prediction

Language Modeling for AAC

- A language model is used to generate the predictions

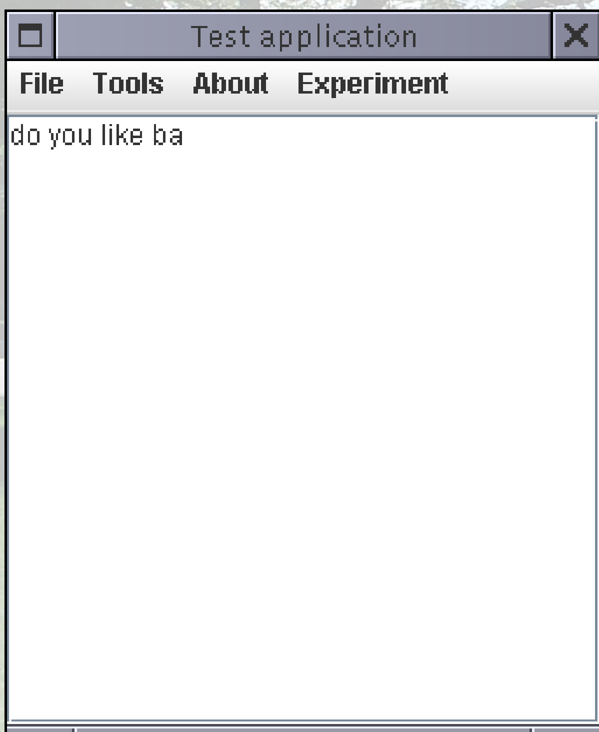


Word	P(w | h)
that	0.01000
a	0.00900
the	0.00850
but	0.00700
be	0.00400
green	0.00390
because	0.00398
ice	0.00350
been	0.00320
back	0.00310
bad	0.00301
basically	0.00270
baby	0.00250
baseball	0.00240

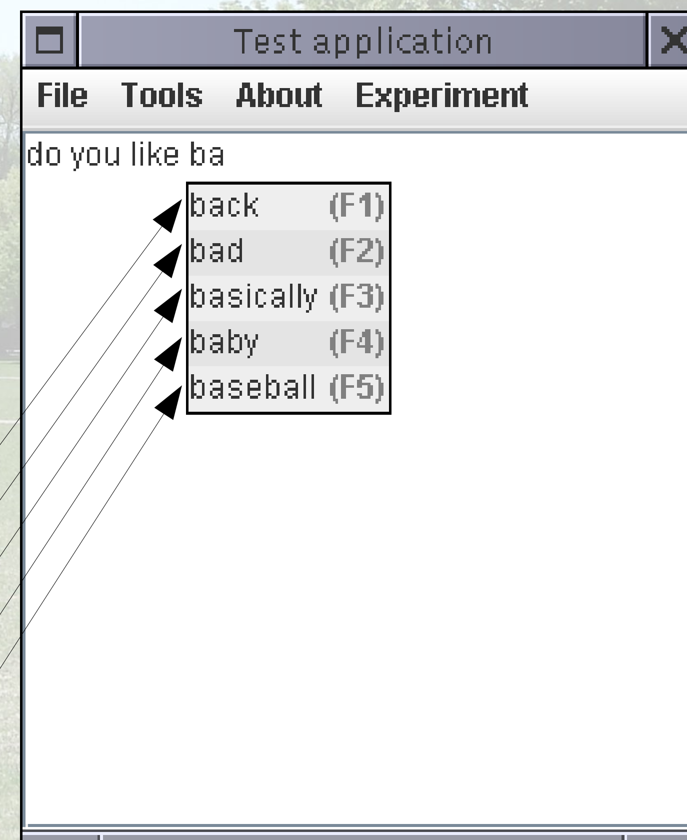


Language Modeling for AAC

- The user presses 'a'



Word	P(w | h)
that	0.01000
a	0.00900
the	0.00850
but	0.00700
be	0.00400
green	0.00390
because	0.00398
ice	0.00350
back	0.00310
bad	0.00301
basically	0.00270
baby	0.00250
baseball	0.00240



Language Modeling for AAC

- Tradition – unigrams and recency/cache
  - Low overlap between the NLP community and AAC community
- Language modeling baseline – trigrams with backoff

Project Goals

- To improve AAC devices by improving the language modeling used in fringe word prediction
- To increase the communication rate of AAC users given a constant rate of input

Evaluation: Keystroke Savings

- Formula

$$KS = \frac{keys_{orig} - keys_{\text{with prediction}}}{keys_{orig}} \times 100\%$$

- Issues
  - Do spaces count?
  - Does pressing enter count?
  - How many predictions?
  - Predict words before a letter is pressed or not? (delayed vs. immediate)

Evaluation: Keystroke Savings

- User simulation
  - A simulated user runs through the software typing the conversation using the fewest number of keystrokes possible
- User interface simulation
  - A space is automatically entered when selecting a predicted word
  - The user can't backspace
- Fringe words only – can't fight “the man” yet

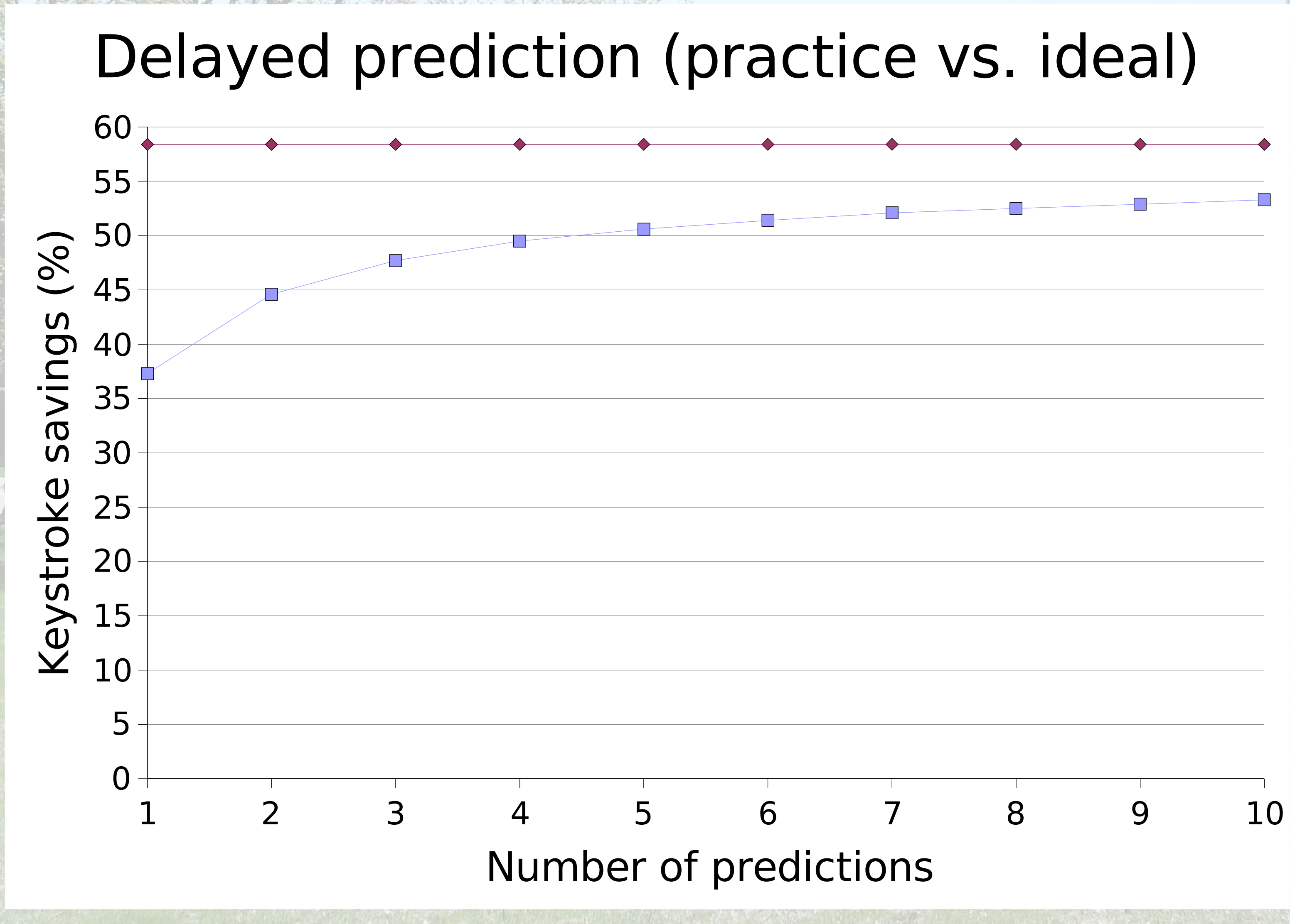
Corpus

- Need a large collection of AAC user text
  - Doesn't exist
- AAC text is conversational, and is thus closer to spoken than written
- Switchboard
  - Telephone conversations (transcribed)
  - ~3 million words, ~2,500 conversations
  - Preprocessing/cleanup example
    - Before: is there um an- is there [cough] a code of dress
    - After: is there a code of dress

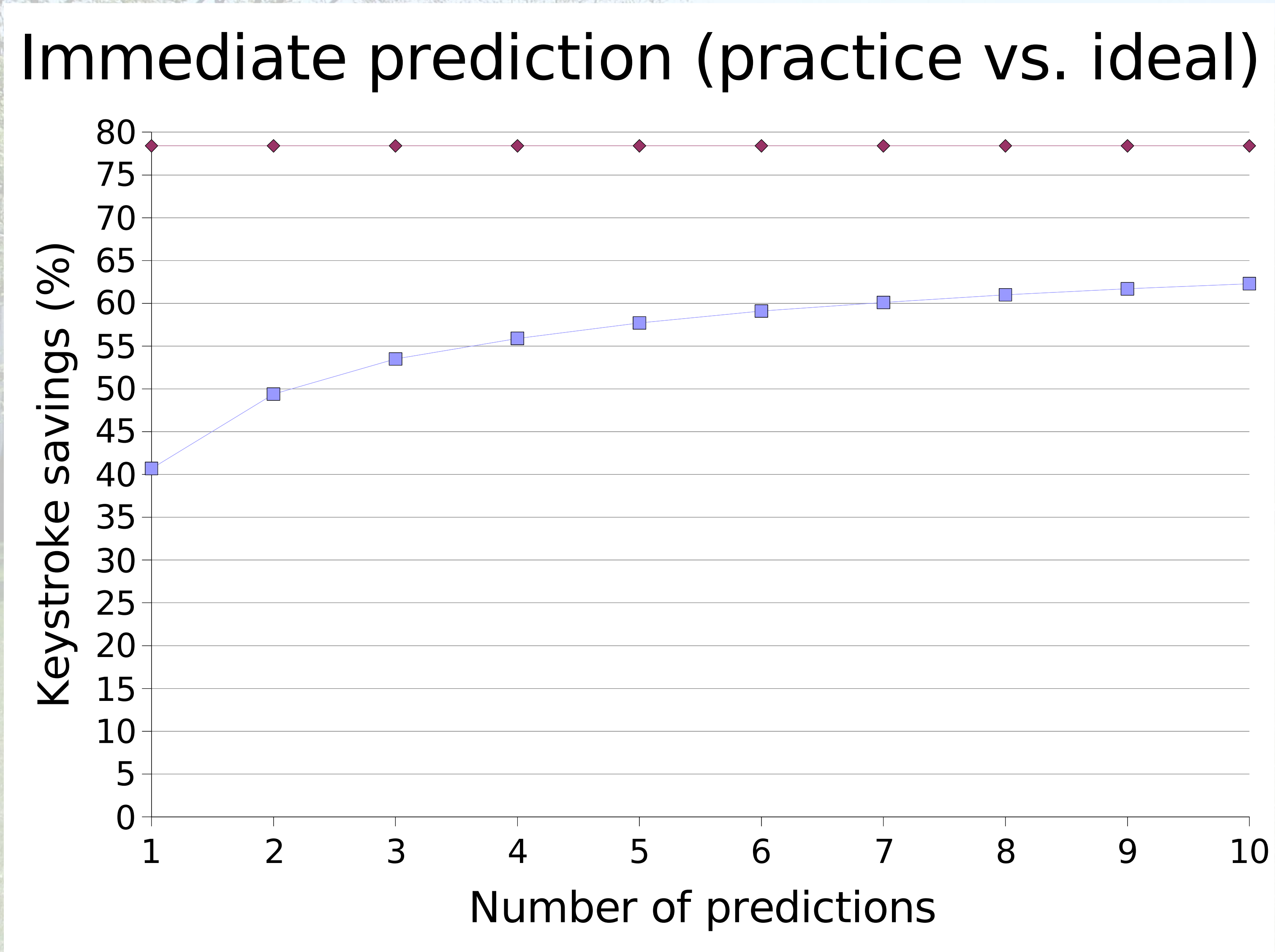
Evaluation – Limits

- Assumption: only single words are predicted
- There is a minimum amount of input required
- ***Delayed prediction*** – requires the first letter be pressed, plus one key to select the word (ideally)
- ***Immediate prediction*** – requires one key to select the word (ideally)
- Both require one key per utterance

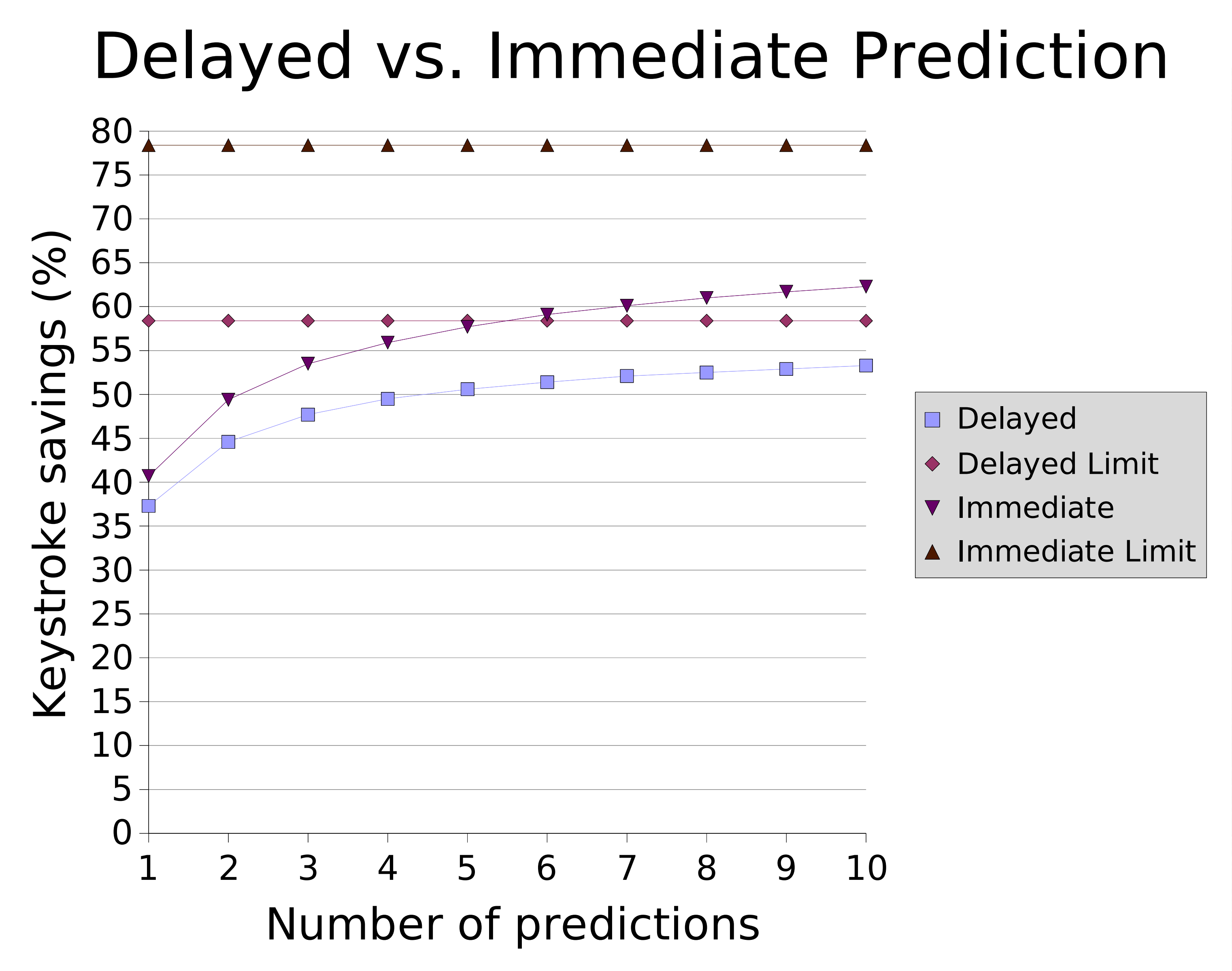
Evaluation – Limits



Evaluation – Limits



Evaluation – Limits



Topic Modeling

- Goal – adapt a language model to the topic of conversation
  - Boost probabilities of on-topic words
  - Depress probabilities of off-topic words
- Overview
  - Obtain a corpus segmented by topic*
  - Determine the topic of conversation based on what has been said so far
  - Create a language model for the current topic

Topic Representation

- In training – a collection of text, split by topic

Switchboard

Topic 1
dress code

wear
shirt
pants
suit
tie
jeans

Topic 2
air pollution

smog
Los Angeles
cars
coal
aerosol

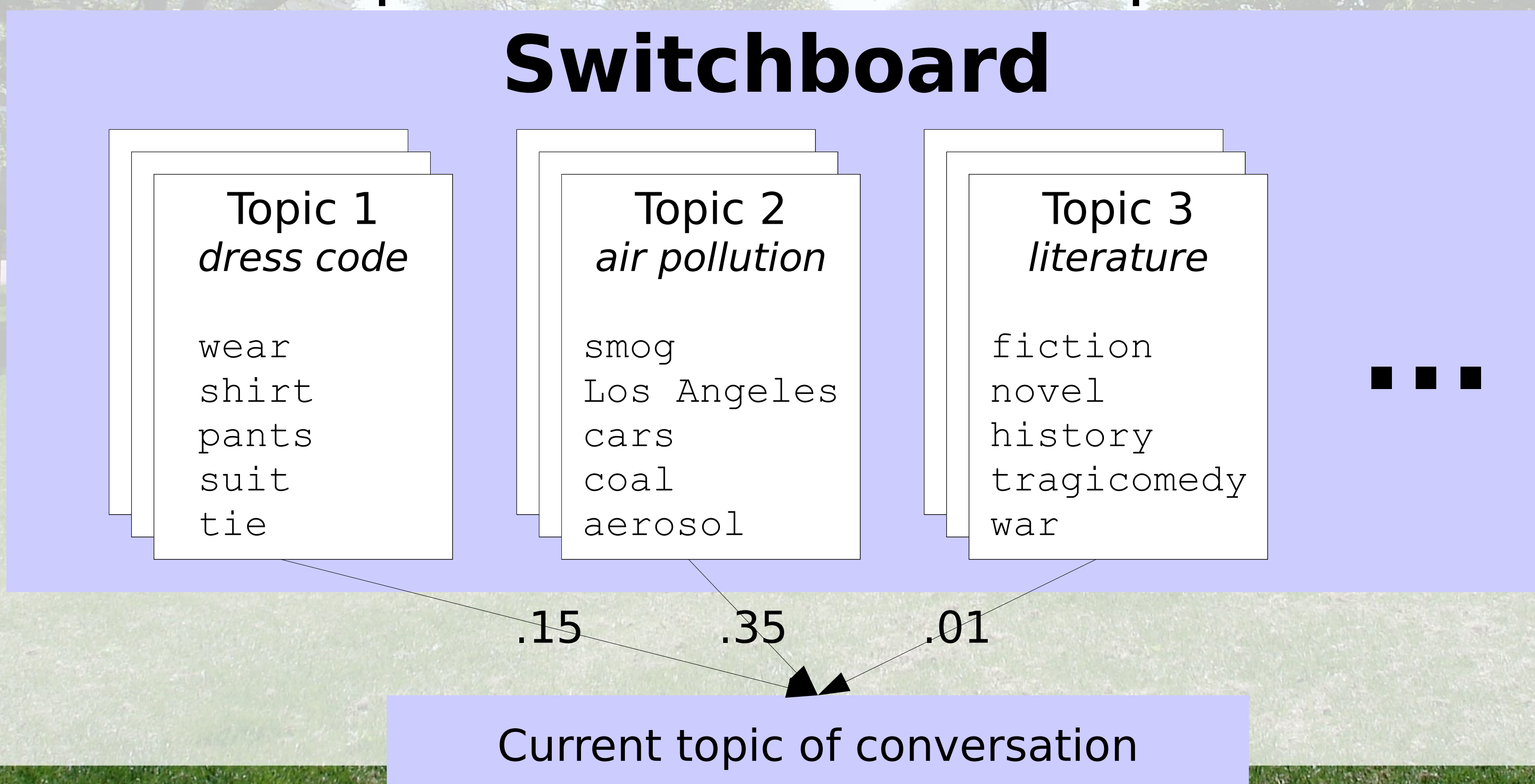
Topic 3
literature

fiction
novel
history
tragicomedy
war
romance

...

Topic Representation

- In testing – a mapping of training topics to weights
  - the compositional nature of topics



Topic Identification

- Cache representation
  - TF-IDF values
  - Exponential decay
  - Words with high IDF excluded (in 85%+ of documents)

Topic Identification

Conversation 2001 – in progress

B: okay hi
A: hi yeah i'd like to talk about how you <u>dress</u> for <u>work</u> and what do you normally what type of <u>outfit</u> do you normally have to <u>wear</u>
B: well i work in <u>corporate control</u> so we have to <u>dress</u> kind of nice so i usually <u>wear</u> <u>skirts</u> and <u>sweaters</u> in the <u>winter time</u> <u>slacks</u> i guess and in the <u>summer</u> just <u>dresses</u>
A: um-hum

Topic Identification

Conversation 2001 – in progress

B: okay hi
A: hi yeah i'd like to talk about how you <u>dress</u> for <u>work</u> and what do you normally what type of <u>outfit</u> do you normally have to <u>wear</u>
B: well i work in <u>corporate</u> <u>control</u> so we have to <u>dress</u> kind of nice so i usually <u>wear</u> <u>skirts</u> and <u>sweaters</u> in the <u>winter</u> <u>time</u> <u>slacks</u> i guess and in the <u>summer</u> just <u>dresses</u>
A: um-hum

Word	Weight
sweaters	2.59
slacks	2.34
skirts	2.33
dresses	2.30
dress	1.85
outfit	1.70
hi	1.69
wear	1.09
corporate	1.00
winter	0.54
normally	0.47
summer	0.22
control	0.17

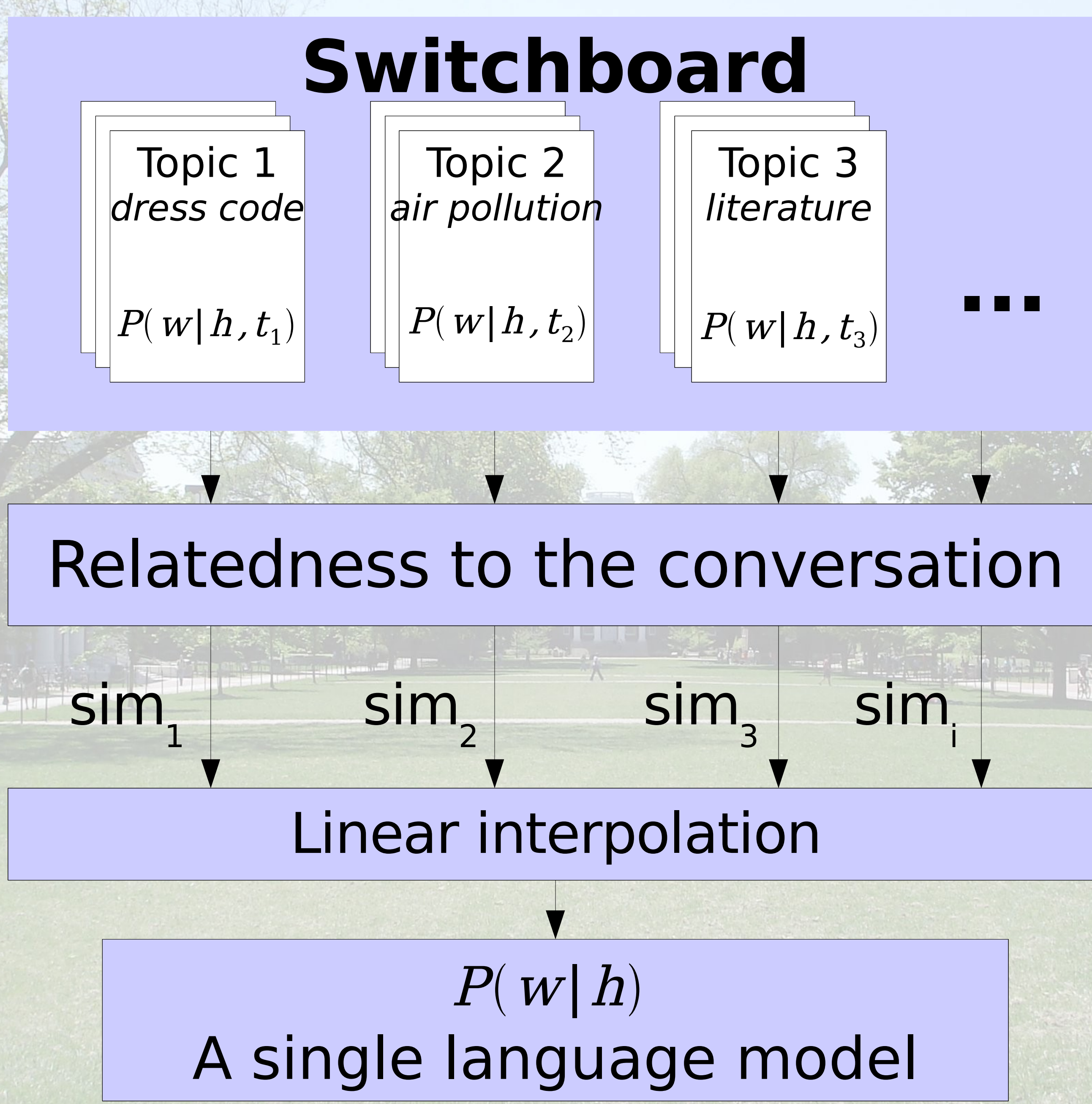
Topic Identification

- Similarity scores
  - Compare the cache and unigram models from the topics
  - Cosine similarity

Topic Application

- Similarity scores
  - Measure a topic's contribution to the final language model
- Linear interpolation of each topic's language model
- Approximate equation

$$P(w|h)=\sum_{i\in topics} P(t_i|h)\times P(w|h,t_i)$$


Switchboard
Topic 1
dress code
$P(w|h,t_1)$
Topic 2
air pollution
$P(w|h,t_2)$
Topic 3
literature
$P(w|h,t_3)$
...
Relatedness to the conversation
sim_1
sim_2
sim_3
sim_i
Linear interpolation
$P(w|h)$
A single language model

Practical Issues

- Re-interpolating the language model can be slow
  - Solution: recompute the model less often, perform smoothing of bigrams on-demand
- Interpolation of frequencies allows for optimization
  - Interpolate frequencies, perform smoothing on-demand
- Smoothing and interpolating
  - Re-scale interpolated frequencies

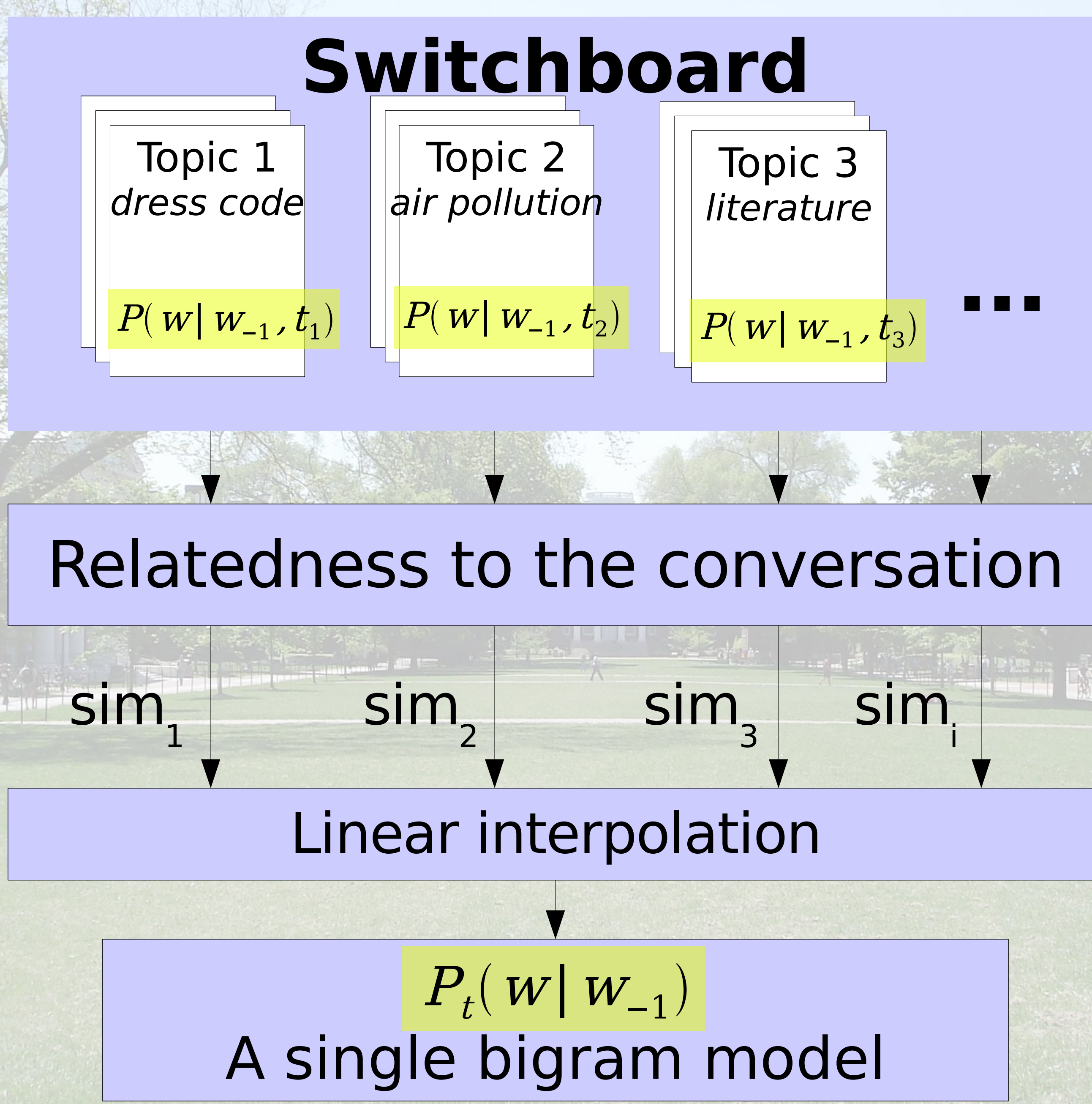
One method, two implementations

- Each topic model has a full-fledged ngram model
  - For computational reasons, bigrams
  - Method A
- Each topic model has a unigram model
  - Needs to be combined with a topic-independent context-aware model (trigrams)
  - Method B

Method A

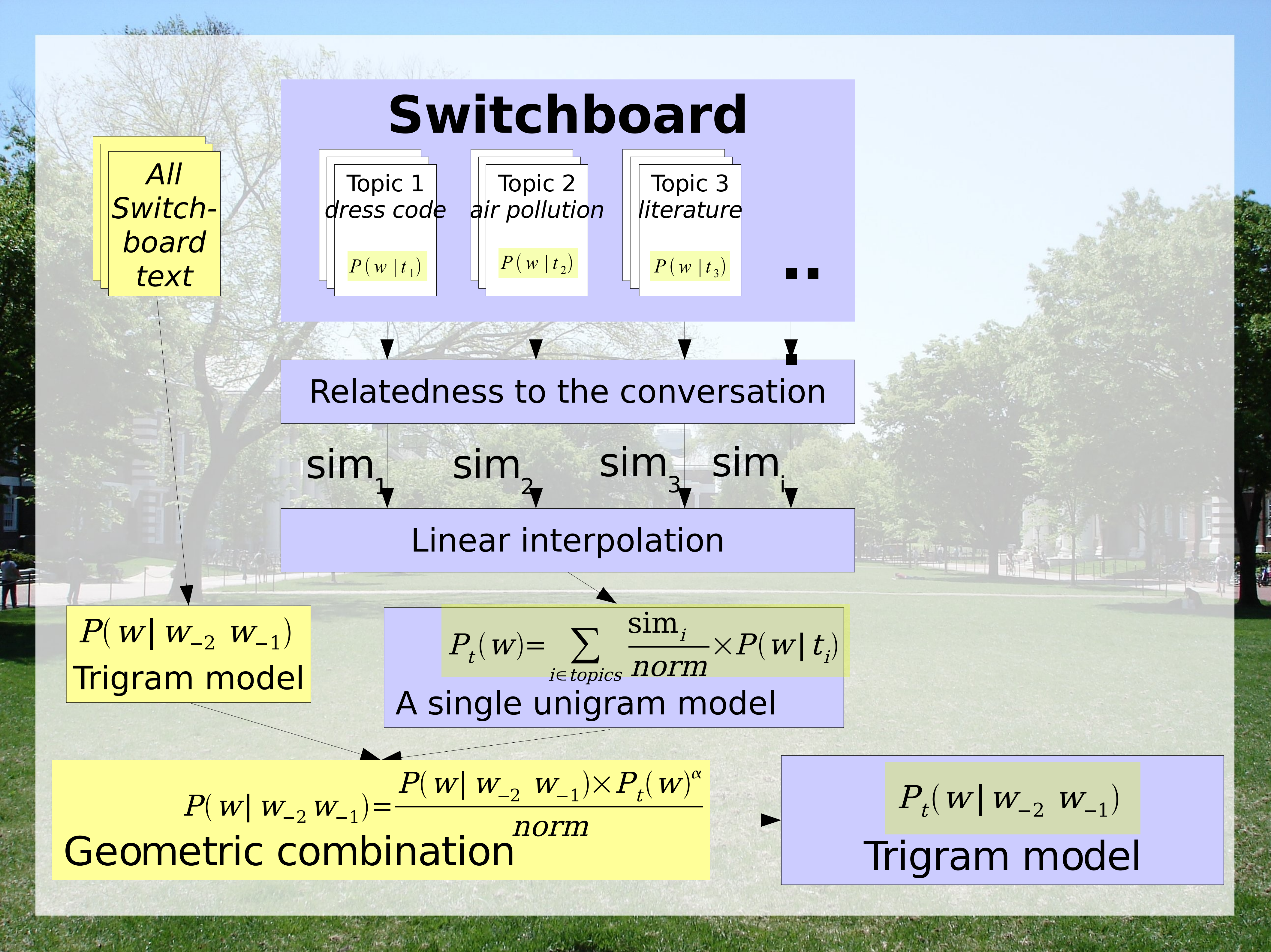
- Each topic model is a bigram model
- Frequencies are interpolated and smoothed to probabilities on-demand
- Approximate equation

$$P(w|w_{-1}) = \sum_{i \in topics} \frac{\mathrm{sim}_i}{norm} \times P(w|w_{-1,}\, t_i)$$


Switchboard
Topic 1
dress code
$P(w|w_{-1}, t_1)$
Topic 2
air pollution
$P(w|w_{-1}, t_2)$
Topic 3
literature
$P(w|w_{-1}, t_3)$
...
Relatedness to the conversation
sim_1
sim_2
sim_3
sim_i
Linear interpolation
$P_t(w|w_{-1})$
A single bigram model

Method B

- Each topic model is a unigram model
- Frequencies are interpolated and smoothed when topic similarity is computed
- Geometric combination of topic-dependent and topic-independent parts
  - Following Bellegarda
  - Exponential weight on the topic component, hand-tuned to about 0.15


Switchboard
Topic 1
dress code
$P(w \mid t_1)$
Topic 2
air pollution
$P(w \mid t_2)$
Topic 3
literature
$P(w \mid t_3)$
..
All Switch-board text
Relatedness to the conversation
sim_1
sim_2
sim_3
sim_i
Linear interpolation
$P(w \mid w_{-2}\ w_{-1})$
Trigram model
$P_t(w) = \sum_{i \in topics} \frac{\text{sim}_i}{norm} \times P(w \mid t_i)$
A single unigram model
$P(w \mid w_{-2}\, w_{-1}) = \frac{P(w \mid w_{-2}\ w_{-1}) \times P_t(w)^{\alpha}}{norm}$
Geometric combination
$P_t(w \mid w_{-2}\ w_{-1})$
Trigram model

Results

- Method A vs. Baseline

Window size	Trigrams	Topic A	Improvement
1	42.3	43.1	0.8
2	51.1	52.3	1.2
3	55.1	56.4	1.3
4	57.3	58.7	1.4
5	58.8	60.2	1.4
6	60.0	61.4	1.4
7	60.8	62.2	1.4
8	61.5	62.9	1.4
9	62.0	63.5	1.5
10	62.5	64.0	1.5

Results

- Method B vs. Baseline

Window size	Trigrams	Topic B	Improvement
1	42.3	42.5	0.2
2	51.1	51.4	0.3
3	55.1	55.4	0.3
4	57.3	57.7	0.4
5	58.8	59.1	0.3
6	60.0	60.3	0.3
7	60.8	61.1	0.3
8	61.5	61.8	0.3
9	62.0	62.3	0.3
10	62.5	62.8	0.3

Results

- Method A vs. Method B

Window size	Topic A	Topic B	Difference
1	43.1	42.5	0.6
2	52.3	51.4	0.9
3	56.4	55.4	1.0
4	58.7	57.7	1.0
5	60.2	59.1	1.1
6	61.4	60.3	1.1
7	62.2	61.1	1.1
8	62.9	61.8	1.1
9	63.5	62.3	1.2
10	64.0	62.8	1.2

Results

- Approximate runtimes
  - Trigram baseline
    1,325 wpm
  - Method A
    32 wpm
  - Method B
    1,267 wpm

Future Work

- Apply the topic modeling methodology to other things
  - Style modeling
    - Grammatical preferences
  - User modeling
    - Affects vocabulary use as well as grammar use
  - Geographical modeling
    - Nearest-neighbor approach
  - Time of day modeling
    - Nearest-neighbor approach

Future Work

- Other language model improvements
  - Better combination of language models
    - Combining recency and trigger-word models with an ngram model
    - Combining selectional restrictions with an ngram model
- Multi-word prediction

Future Work

- Better evaluation
  - Evaluate word prediction not just for one type of conversation, but many types of communication that AAC users do
    - Pre-planned speech
    - Spontaneous speech (like Switchboard, SBCSAE, etc)
    - Papers
    - Emails
  - Cross-domain testing to estimate real-world performance

Current Work

- User studies – will better word prediction help AAC users significantly?

Conclusions

- NLP techniques can be applied to improve AAC devices.
- Bigram and trigram models predict words fairly well.
- Topic modeling improves the prediction of fringe words somewhat.
- Other language modeling improvements are likely to improve keystroke savings in word prediction.

Acknowledgments

- This work has been supported by the U.S. Department of Education Grant number: H133G040051, Field Initiated Development Project from the National Institutes on Disability and Rehabilitation Research.