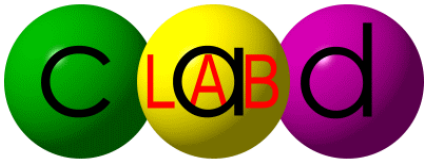




# Non-Syntactic Word Prediction for AAC

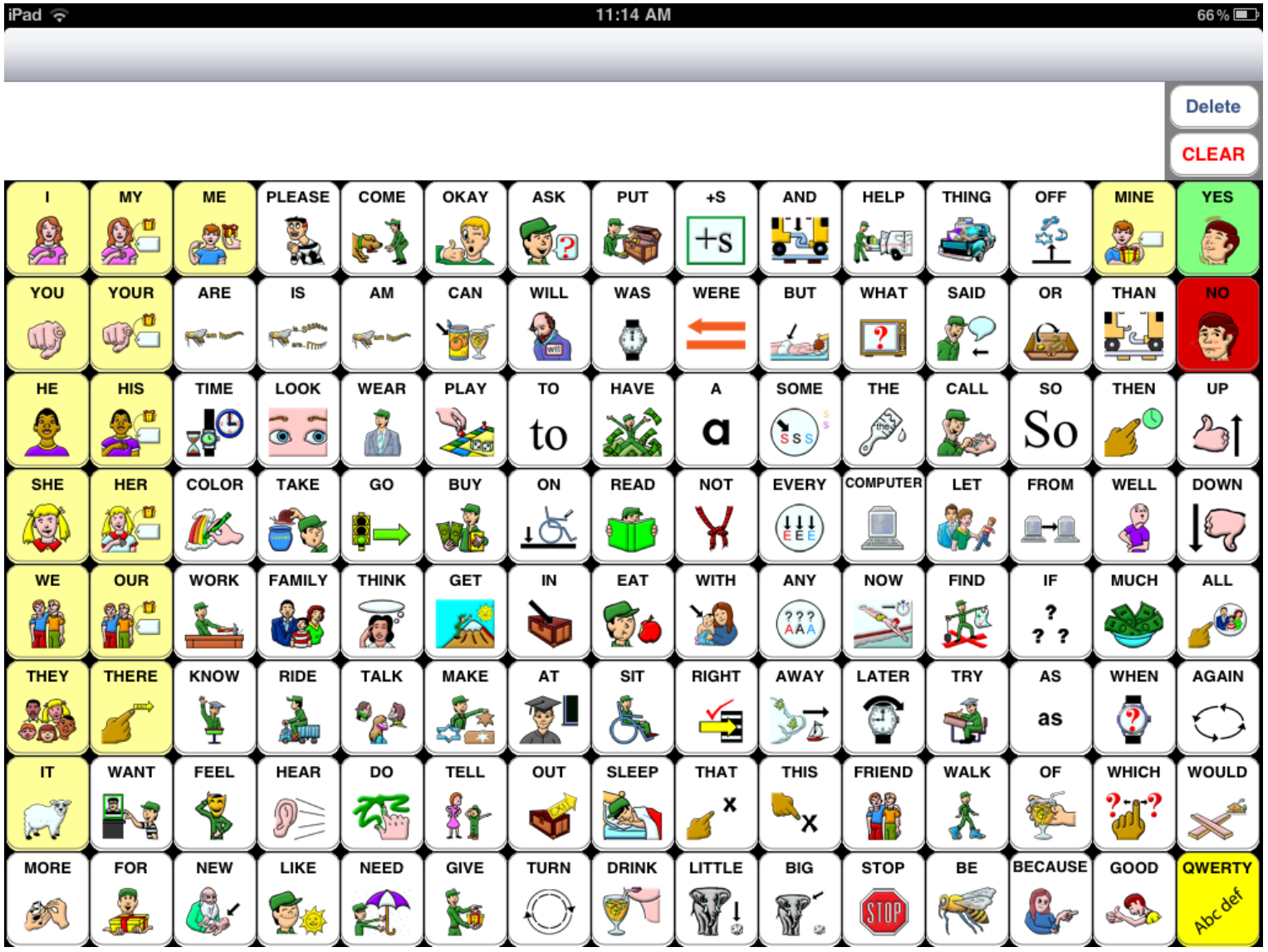
Karl Wiegand
Northeastern University
Boston, MA USA

Rupal Patel, Ph.D.
Northeastern University
Boston, MA USA


This work is supported by the National Science Foundation under Grant No. 0914808.

# Target Systems: Example 1

# Target Systems: Example 2



# Background

- Based on written language
- Users currently select letters, words, or icons in syntactically correct order*
- Non-syntactic input usually results in non-syntactic output

*H. Van Balkom and M. Welle Donker-Gimbrere. 1996. A psycholinguistic approach to graphic language use. Augmentative and alternative communication: European Perspectives, pages 153–170.

# Observations

- Speed of communication is important
- Complete/correct utterances are important
- Language style may be important
- There are existing strategies for:
  - Completing words
  - Completing syntactic utterances
  - Detecting missing letters of words

# Prior and Related Work

- Missing function words (Compansion)
  - McCoy et al., 1998
- Missing content words (memory-based language models using trigrams)
  - Van Den Bosch et al., 2006 and 2009
- Word relationships and disambiguation based on grammatical characteristics (IR)
  - Tzoukermann et al., 1997; Allan and Raghavan, 2002

# Prior and Related Work

- Word relationships based on semantic characteristics and roles (IR)
  - Westerman and Cribbin, 2000; Fang and Zhai, 2006; Hemayati et al., 2007
- Word relationships based on distance and collocation (IR)
  - Lin and Hovy, 2003; Lv and Zhai, 2009
  - Matiasek and Baroni, 2003 (moving window)
  - Jarvelin et al., 2007 (s-grams)

# Motivation

- Current completion and prediction strategies rely on syntactic input and word distance
- N-gram statistics are widely available for well-ordered input
- If the input isn't syntactically correct or well-ordered, can complete utterances still be predicted?

# Exemplar

"I like to play chess with my brother."
"My brother and I play video games."
"I play chess with my dad."

Input: like, play, chess, i, brother
Input: play, video games, i, brother
Input: play, chess, i, dad
Input: i, brother, ...

How can we track these relationships?

# Possible approach

- Sentences are one of the smallest units of language that are:

  - Semantically coherent
  - Semantically cohesive
  - Syntactically demarcated

  ... could be leveraged for prediction ...

# Semantic Grams

- A multiset of words that appear together in the same sentence.

Sentence: "I like to play chess with my brother."

| | |
|---|---|
| brother, chess (1) | brother, i (1) |
| brother, like (1) | brother, play (1) |
| chess, i (1) | chess, like (1) |
| chess, play (1) | i, like (1) |
| i, play (1) | like, play (1) |

# More on Sem-grams

- Sentence Boundary Detection is fast and relatively accurate (> 98.5%)
- Sentence-level co-occurrence with uniform weight applied to all relationships in a sentence
- Order-independent and no null elements

# Technical Problem Definition

Given:

- Multiset of existing words E
- Set of candidate words C

Output:

- Most likely (argmax) candidate word $c \in C$

  - or -

- Ranked list of candidate words $c \in C$

# Four Prediction Algorithms

- S1: Conditional independence of existing words to each other (naive Bayes)
- S2: Random drawing of sem-grams from an existing pool
- N1: Copy of S1, but with unordered n-grams; reward adjacency next to all existing words
- N2: Reward adjacency next to at least one existing word (strength of n-grams?)

# Corpus

- Blog Authorship Corpus
  - 140 million words
  - 19,320 bloggers in August 2004
  - Age range of 13 - 48
  - Equally divided between males and females
- Pre-processing:
  - Split sentences and words
  - Remove stop words
  - Stem words
  - Check stems for dictionary membership
- Split by authors: 80% training, 20% testing
- Plus-one smoothing on trained sem-grams and n-grams (bigrams)

# Method

For every test sentence:

1. Process (split, stop, stem, and check)
2. Shuffle stems
3. Remove one (target)
4. Ask each algorithm to predict the missing stem by providing a ranked list of guesses

Evaluation (random 2000 sentences):

- Score = position of target word; lower scores are better

# Method

- Test sentences truncated to 20 words
- N-gram algorithms seeded with top 10 unordered n-grams for each input word
- Sem-gram algorithms seeded with top 10 sem-grams for each input word
- Maximum of 190 candidate words to rank
- Ranked lists truncated to 100; otherwise, considered a "failure to predict"

# Results: Sample 1

**Original Sentence:**

"but i went to church yesterday with the fam."

**Target Stem:** went

**Input Stems:** yesterday, church

**N1 Candidate List:** went, morn, today, go, attend, work, afternoon, church, got, day, ...

**S1 Candidate List:** went, go, church, today, got, day, like, time, just, well, one, get, peopl, ...

# Results: Sample 2

**Original Sentence:**

"This semester Im taking six classes."

**Target Stem:** class

**Input Stems:** take, semest, six

**N1 Candidate List:** next, month, class, hour, last, second, week, year, first, five, flag, ...

**S1 Candidate List:** class, month, year, last, time, one, go, day, get, school, will, first, ...

# Results: Sample 3

**Original Sentence:**

> “Hey, they’re in first, by a game and a half over the Yankees.”

**Target Stem:** game

**Input Stems:** yanke, hey, first, half

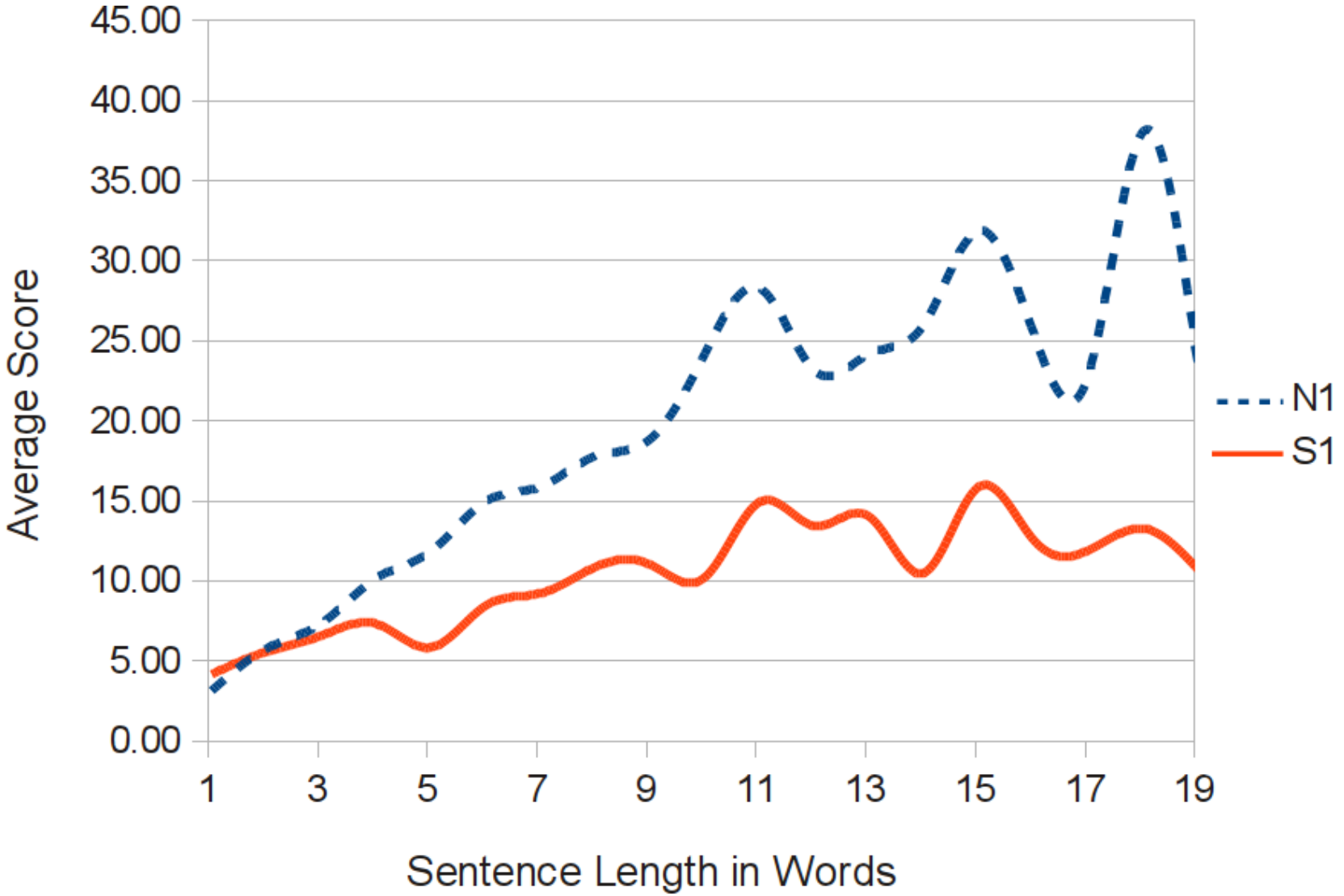
**N1 Candidate List:** game, stadium, like, hour, time, year, day, guy, hey, fan, say, one, two, ...

**S1 Candidate List:** game, got, like, red, time, play, team, sox, hour, go, fan, one, get, day, ...

# Summary of Results

| | N1 | N2 | S1 | S2 |
|---|---|---|---|---|
| **# of Sentences** | 2000 | 2000 | 2000 | 2000 |
| **# Predicted** | 647 | 649 | 435 | 435 |
| **Average Score** | 16.26 | 19.70 | 9.04 | 12.67 |

# Results by Sentence Length



# Issues and Future Directions

- Accuracy vs. Coverage
- Use of bigrams
- Seeding the candidate list
- Computational requirements

- Hybrid approaches:
  - Identical seed lists
  - Smooth from n-gram prediction to sem-gram prediction based on sentence length
  - Merge prediction lists

- BAC provides age, gender, and occupation

# Application: Single-Page App

# Application: Multi-Page App

# Summary

- Unordered/non-syntactic prediction is possible
- N-grams can provide broad coverage
- Sem-grams may provide better accuracy
- Sem-grams may inform system behavior

Thank you!