[1]Department of Linguistics, [2]Department of Computer Science and Engineering, [3]Department of Family Medicine

# Interpreting Questions with a Log-Linear Ranking Model in a Virtual Patient Dialogue System

Evan Jaffe[1], Michael White[1], William Schuler[1], Eric Fosler-Lussier[2], Alex Rosenfeld, Douglas Danforth[3]

## Introduction

Objective: Train medical students using virtual standardized patients (VSPs)

Current Approach: ChatScript pattern matching engine

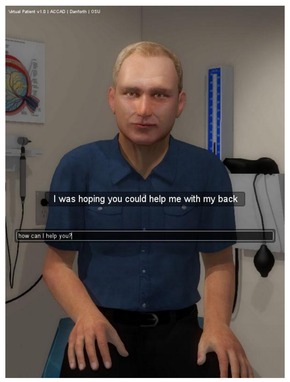
Problems: Low accuracy, authoring burden, no confidence measure

Proposed Solution: Log-linear ranking model is data-driven and provides a confidence measure

## Background and Related Work

- Paraphrase Identification - Microsoft Research Paraphrase Corpus (Dolan et al, 2004)
  - Binary classifier vs. Ranking (Ravichandran et al. 2003)
  - Strong Lexical Overlap baseline (Das and Smith, 2009)
- Classification
  - Maxent multiclass classifier (DeVault et al., 2011)
  - Current ChatScript system

**Fig 1: Example Exam Room and Virtual Patient Avatar**




## The Model

**Eq 1: Probability of a class given an input sentence**

$$P(c|x) = \frac{1}{Z(x)} \sum_{v \in c} \exp\Big( \sum_j w_j f_j(x, v) \Big)$$

*Eq 1: Probability of a class c given an input question x is the normalized sum over variants v of the class of the exponentiated dot product of the weight and feature vectors, w and f, respectively.*

## The Model, cont.

**Eq 2: Training Objective**

$$\sum_i \log P(c_i|x_i) - \lambda \sum_j w_j^2$$

*Eq 2: Training objective is to find feature weights that maximize the regularized log likelihood of the canonical question c given input sentence x, minus a Gaussian prior regularization term. We used Hal Daumé III's MEGAM.*

**Eq 3: Test Objective**

$$c^* = c(v^*), \text{where}$$
$$v^* = \text{argmax}_v \sum_j w_j f_j(x, v)$$

*Eq 3: Test objective is to choose the class c*, where the canonical question or any of its variants counts as a correct answer. The variable v* is the closest variant to test sentence x, which is the argmax over variants of the dot product of the weight and feature vectors.*

## Features

- Align - Meteor alignment overall score
- Lexical Overlap
  - 1-3 gram precision/recall exact/stem n-gram matching
- Lex
  - Binary indicator features for matching or failing to match a given word
- Weighting
  - IDF weighting (canonical plus its variants as a document)
  - Corpus frequency weighting (negative log probability)
- Concept
  - 1-2 gram precision/recall lexical overlap matching that substitutes words or phrases for their matching 'concept' (hand-crafted hypernym)

concept: **~medicines** [ ~drugs legal analgesia antibiotics antidote claritin drug drugs hormone hormonal loratidine medication medications **medicine** meds narcotic 'pain killer' 'pain killers' painkiller pill prescription 'prescription medication' 'prescription medications' remedy steroid **tablet** tums]

What **kind** of **medicine** is that?

What **type** of **tablet** would that be?

Unweighted exact unigram precision = 3/6 = 0.5
Unweighted exact unigram recall = 3/7 = 0.43
Unweighted concept unigram precision = 5/6 = 0.83
Unweighted concept unigram recall = 5/7 = 0.71

## Interpretation Experiment

- 32 dialogues, 918 user turns, mean 29 turns per dialogue
- Asked question, canonical question, current topic and question response are annotated for each turn
- 193 canonical questions
- 787 question variants, mean 4.1 variants per canonical question
- Feature subsets generate a number of models, accuracy shown below

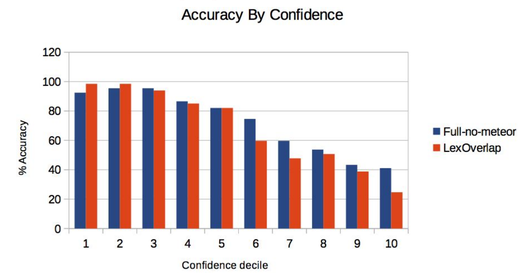
**Table 1: Accuracy by model**

| Model Name | Features Included | % Accuracy |
|---|---|---|
| Align | Meteor score feature alone | 75.3 |
| LexOverlap | Das and Smith-style lexical overlap baseline | 74.9 |
| LexOverlap+lex | adds lexical features | 74.1 |
| LexOverlap+align | adds Meteor score feature | 75.8 |
| LexOverlap+weighting | adds weighting features | 77.8 |
| LexOverlap+concept | adds concept features | 78.1 |
| LexOverlap+concept+ weighting | adds weighting and concept features | 78.5 |
| Full | all features | 77.0 |
| Full-no-meteor | full minus Align and Meteor features | 78.6 |

**Eq 4: Calculating Confidence**

$$P(v|x) = \frac{\exp \sum_j w_j f_j(x, v)}{\sum_v \exp \sum_j w_j f_j(x, v)}$$

**Chart 1: Accuracy by confidence**




## Conclusions

- Log-linear ranking model (~78%) outperforms DeVault-style multiclass classifier (~67%)
- Concept features most useful addition
- Confidence measure correlates with accuracy

## Further Study

- Collect larger training corpus (100 dialogue set, 5000 user turns in progress)
- Robustness to noisy ASR input
- Vector-space models of word meaning to better identify paraphrases

## Acknowledgements

*Thanks to Kellen Maicher for creating the virtual environment, Bruce Wilcox for authoring ChatScript, and Laura Zimmerman for managing the laboratory and organizing student involvement.*

*This project was supported by funding from the Department of Health and Human Services Health Resources and Services Administration (HRSA D56HP020687) and the National Board of Medical Examiners Edward J. Stemmler Education Research Fund (NBME 1112-064).*

## Contact Information

*Evan Jaffe: jaffe.59@osu.edu*

*Michael White: white.1240@osu.edu*