Lexical Cognitive Networks with Application to Semantic Similarity Computation and Affective Text Analysis

Alexandros Potamianos

Dept. of ECE, Technical Univ. of Crete, Chania, Greece
Acknowledgements

- Elias Iosif: Semantic similarity computation, semantic networks
- Nikos Malandrakis: Affective models for text and multimedia
- Shri Narayanan (USC): Affective modeling of dialogue interaction

References

Problem Definition

- Semantic Similarity Computation
  - Given a pair of words or terms \((w_i, w_j)\)
  - Compute semantic similarity between them \(S(i, j)\)

- Related tasks
  - Phrase or sentence level semantic similarity
  - Strength of associative relation between words
  - Affective score (valence) of words and sentences

- Motivation
  - Organizing principle of human cognition
  - Building block of machine learning in NLP/semantic web
  - Entry point for the semantics of language
Problem Definition

- How to define **semantic** relations between words?
  - *Linguists*: well-defined in terms of a handful of relations
  - *Cognitive scientists*: no idea!

- How to define **associative** relations between words?
  - *Linguists*: long-tail of relations, why is this thing useful anyhow?
  - *Cognitive scientists*: well-defined experimentally via **priming**

- What is **lexical priming**?
  - Activation of associated words in cognitive lexical net
  - Cache for quick access of most probable candidates
  - Look-ahead useful for pruning improbable hypotheses
System 1 vs System 2

- Using Kahneman’s (and others) formalism:
  - System 1 (intuition): generates
    - impressions, feelings, and inclinations
  - System 2 (reason): turns System 1 input into
    - beliefs, attitudes, and intentions

- Associative relations reside in System 1
- But where do semantic relations reside?
Example

Example from vision: system 1 vs system 2
Semantic Relationship Scoring

- **Experiment**
  - Please **quickly** rate the following word pair in terms of **semantic similarity** using a score between 0 (totally dissimilar) and 4 (semantically equivalent). Record this score.
  - Then **take your time adjusting this score** to the most appropriate semantic similarity value between 0 and 4.

**Word pair**: (hand, glove)

**Results**:
- System 1: strong association fast score 3 or 4
- System 2: weak (or non-existent) semantic relationship slow score 2 or 3
Semantic Relationship Scoring

- **Experiment**
  - Please **quickly** rate the following word pair in terms of **semantic similarity** using a score between **0** (totally dissimilar) and **4** (semantically equivalent). Record this score.
  - Then **take your time adjusting this score** to the most appropriate semantic similarity value between 0 and 4.

- **Word pair**: *(hand, glove)*
Semantic Relationship Scoring

- **Experiment**
  - Please quickly rate the following word pair in terms of **semantic similarity** using a score between 0 (totally dissimilar) and 4 (semantically equivalent). Record this score.
  - Then take your time adjusting this score to the most appropriate semantic similarity value between 0 and 4.

- **Word pair**: (hand, glove)

- **Results**:
  - System 1: strong association **fast score 3 or 4**
  - System 2: weak (or non-existent) semantic relationship **slow score 2 or 3**
Associative Anchoring

- What you experienced: **associative anchoring**
- **Anchoring** is a cognitive deficiency due to system 1 vs 2 cognitive organization, e.g.,
  - \( x = 1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \) vs \( y = 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 \)
  - **Fast** estimate of \( y \) greater than that of \( x \)
  - **Slow** estimate of \( y \) greater than that of \( x \)
- Semantic score of (hand, glove) should be 0 or 1
- instead due to associative anchoring 2 or 3

Semantic score a (system 2) **post-correction** of association (system 1) score
A cognitive model of lexical semantics

- **Distributed** representation [*Rogers & McClelland, ’04*]
- **Similarity:** common vs. distinctive attributes [*Tversky, ’77*]
Main approaches of lexical semantics

- Word are associated with **feature** vectors
  - crisp, parsimonious representation of semantics
- Distributional semantic models (DSMs)
  - Semantic information extracted from word frequencies
  - Estimate **co-occurrence counts** of word pairs or triplets
  - Estimate statistics of **word context** vectors
- Semantic **networks**
  - discovery of new relations via **systematic co-variation**
  - **robust** estimates – smoothing corpus statistics over network
  - rapid language acquisition
Example of Semantic Network

- **Linked nodes**: lexicalized *senses* and *attributes*
- Informative for *semantic similarity* computation
- Computation of *structural* properties, e.g., *cliques*
Proposed semantic similarity two-tier system

- Unifies the three approaches
- **Fuzzy** vs explicit semantic relations
- **Word senses** vs **words** vs **concepts**
- A two tier system
  - An **associative** network backbone
  - Semantic relations defined as operations on network neighborhoods (**cliques**)
- Consistent with system 1 vs system 2 view
- Furthermore we believe that the
  - underlying network consists of **word senses**, and
  - is a **low dimensional semi-metric space**
Semantic Similarity Estimation by Machines

- **Resource-based**, e.g., WordNet
  - Require expert knowledge
  - Not available for all languages

- **Corpus-based**
  - Distributional semantic models (DSMs)
  - Unstructured (unsupervised): no use of linguistic structure
  - Structured: use of linguistic structure
  - Pattern-based, e.g., Hearst patterns

- **Mixed**
Semantic Sim. Computation: Sense Similarity

- **Maximum sense similarity assumption** [Resnik, ’95]:
  - Similarity of words equal to similarity of their closest senses
  - If words are considered as sets of word senses, this is the “common sense” set distance

- Given words $w_1, w_2$ with senses $s_{1i}, s_{2j}$

  $$S(w_1, w_2) = \max_{ij} S(s_{1i}, s_{2j})$$
Semantic Sim. Computation: Attributional Similarity

Attributional similarity assumption

- Attributes (features) reflect semantics
  - Item-Relation-Attribute, e.g., canary-color-yellow

- Main representation schemes
  - Hierarchical/Categorical
    - Mainly taxonomic relations, e.g., IsA, PartOf
  - Distributed (networks)
    - Open set of relations, e.g., Cause-Effect, etc

- Similarity between words
  - Function of attribute similarity
  - Defined wrt representation
Types of Similarity Metrics

- **Co-occurrence-based**
  - Assumption: co-occurrence implies relatedness
  - Co-occurrence counts: web hits, corpus-based
  - Examples: Dice coef., point-wise mutual information, ...

- **Context-based**
  - Assumption: context similarity implies relatedness
    (distributional hypothesis of meaning)
  - Contextual features extracted from corpus
  - Examples: Kullback-Leibler divergence, cosine similarity, ...

- **Network-based (proposed)**
  - Build lexical net using co-occurrence and/or context sim.
  - Notion of semantic neighborhoods
  - Assumptions: neighborhoods capture word semantics
Queries to Web Search Engines

- Number of hits
- Document URLs (download)
- Document snippets
Corpus Creation using Web Queries

- Two types of web queries
  - AND, e.g., “money + bank”
    “… leading **bank** in India offering online **money** transfer ...”
  - IND, e.g., “bank”
    “… downstream parallel to the **banks** of the river ...”

- AND queries
  - Pros: Similarity computation **highly correlated** (0.88) with human ratings [Iosif & Potamianos, ’10]
  - Cons: **Quadratic** query complexity wrt lexicon $L$

- IND queries
  - Pros: **Linear** query complexity wrt lexicon $L$
  - Cons: **Sense ambiguity**: moderate correlation (0.55)
Semantic Similarity Estimation

- **Co-occurrence** based metrics
  - From web: hits of IND, AND queries
  - From (web) corpus: co-occurrence counts at the snippet or sentence level
  - Metrics: Dice, Jacard, Mutual Information, Google

- **Context-based** metrics
  - Download a corpus of documents of snippets using IND queries
  - Construct lexical context vector for each word (window ±1)
  - Cosine similarity using binary or log-weighted counts
Why do IND queries fail to achieve good performance?

1. Word senses are often semantically diverse
   - co-occurrence acts as a semantic filter
2. Word senses have poor coverage in IND queries
   - rare word senses of words not well-represented

Solution: use semantic networks

1. Create a corpus for all words in lexicon (not just semantic similarity pair)
2. Use semantic neighborhoods for semantic cohesion
   - improved robustness
3. Inverse frequency word-sense discovery
   - discover rare senses via co-occurrence with infrequent words
Corpus and Network Creation

- **Goals**
  - Linear web query complexity for corpus creation
  - New similarity metrics with high performance

- **Proposed method**
  - IND queries to aggregate data for large $L \approx 9K$
  - Create network and semantic neighborhoods
  - Neighborhood-based similarity metrics

- **Advantages**
  - Network: parsimonious representation of corpus statistics
  - Smooth distributions
  - Rare words: well-represented
  - Enable discovery of less frequent senses
Lexical Network - Semantic Neighborhoods

Lexical Network

- Undirected graph $G = (N, E)$
  - Vertices $N$: words in lexicon $L$
  - Edges $E$: word similarities

Semantic Neighborhoods

- For word $i$ create subgraph $G_i$
- Select neighbors of $i$
  - Compute $S(i, j), \forall j \in L, i \neq j$
  - Sort $j$ according to $S(i, j)$
  - Select $|N_i|$ top-ranked $j$
## Semantic Neighborhoods: Examples

<table>
<thead>
<tr>
<th>Word</th>
<th>Neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td>automobile</td>
<td>auto, truck, vehicle, car, engine, bus, ...</td>
</tr>
<tr>
<td>car</td>
<td>truck, vehicle, travel, service, price, industry, ...</td>
</tr>
<tr>
<td>slave</td>
<td>slavery, beggar, nationalism, society, democracy, aristocracy, ...</td>
</tr>
<tr>
<td>journey</td>
<td>trip, holiday, culture, travel, discovery, quest, ...</td>
</tr>
</tbody>
</table>

- **Synonymy**
- **Taxonomic**: IsA, Meronymy
- **Associative**
- **Broader semantics/pragmatics**
- ...
Neighborhood-based Similarity Metrics: $M_n$

$M_n$ metric: maximum similarity of neighborhoods

- Motivated by maximum sense similarity assumption
  - Neighbors are semantic features denoting senses
  - Similarity of two closest senses
- Select max. similarity: $M_n(\text{“forest”}, \text{“fruit”}) = 0.30$
Neighborhood-based Similarity Metrics: $R_n$

$R_n$ metric: correlation of neighborhood similarities

- Motivated by **attributional similarity** assumption
  - Neighborhoods encode word **attributes (or features)**
  - Similar words have **co-varying sim.** wrt their neighbors
- Compute correlation $r$ of neighborhood similarities
  - $r_1((0.16...0.09), (0.10...0.01)), r_2((0.002...0), (0.63...0.13))$
- Select **max. correlation**: $R_n("forest", "fruit") = -0.04$
Neighborhood-based Similarity Metrics: metric $E_n^{\theta=2}$

$E_n^{\theta=2}$ metric: sum of squared neighborhood similarities

Motivation: middle road between $M_n$ and $R_n$
- Accumulation of word-to-neighbor similarities
- Non-linear weighting of similarities via $\theta = 2$

$E_n^{\theta=2}(\text{“forest”}, \text{“fruit”}) = \sqrt{(0.10^2 + \cdots + 0.01^2) + (0.002^2 + \cdots + 0^2)} = 0.22$
Performance of net-based similarity metrics

- **Task**: similarity judgment on noun pairs
- **Dataset**: MC [Miller and Charles, 1998]
- **Evaluation metric**: Pearson’s correlation wrt to human ratings

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Neighbor selection</th>
<th>Similarity computation</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC</td>
<td>co-occur.</td>
<td>co-occur.</td>
<td>$M_{n=100}$</td>
</tr>
<tr>
<td>MC</td>
<td>co-occur.</td>
<td>context</td>
<td>0.91</td>
</tr>
<tr>
<td>MC</td>
<td>context</td>
<td>co-occur.</td>
<td>0.52</td>
</tr>
<tr>
<td>MC</td>
<td>context</td>
<td>context</td>
<td>0.51</td>
</tr>
</tbody>
</table>
Main findings

- **Network** construction
  - Co-occurrence metrics achieve high-recall for *word senses*
  - Context-based metrics achieve high-recall for *attributes*

- Semantic similarity performance
  - Co-occurrence a more *robust* feature than context
  - Max *sense* similarity assumption is valid and gives best performance
  - Attributional similarity assumption valid for certain cases/languages
Performance of web-based similarity metrics

- For **MC** dataset

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>context</td>
<td>AND queries</td>
<td>0.88</td>
</tr>
<tr>
<td>context</td>
<td>IND queries</td>
<td>0.55</td>
</tr>
<tr>
<td>context</td>
<td>IND queries: network</td>
<td>0.90</td>
</tr>
</tbody>
</table>

- **Comparable** to structured DSMs, WordNet-based approaches
Minimum Error Semantic Similarity

- **Assumption 1**
  Senses lexicalized as *single words*

- **Assumption 2**
  Sim. of $w_i$, $w_j$: pairwise max. sim. between their senses

- **Assumption 3**
  3a. $w_i$, $w_j$ always co-occur with their *two closest* senses
  3b. ...

- **Assumption 4**
  4a. Uniform distribution of senses
  4b. ...
Performance of min error semantic similarity

- Modify pointwise mutual info. \( I(w_i, w_j) = \log \frac{\hat{p}(w_i, w_j)}{\hat{p}(w_i) \hat{p}(w_j)} \) as

\[
I_{\alpha}(w_i, w_j) = \frac{1}{2} \left[ \log \frac{\hat{p}(w_i, w_j)}{\hat{p}^\alpha(w_i) \hat{p}(w_j)} + \log \frac{\hat{p}(w_i, w_j)}{\hat{p}(w_i) \hat{p}^\alpha(w_j)} \right]
\]

- Assumptions: 1, 2, 3a, and 4b
- Co-occurrence considered at sentence-level
- \( \alpha \) estimated to max. sense coverage of sem. neigh.
- Task: similarity judgment, correlation wrt to human ratings

<table>
<thead>
<tr>
<th>Dataset</th>
<th>( I )</th>
<th>( I_{\alpha} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC</td>
<td>0.78</td>
<td>0.89</td>
</tr>
<tr>
<td>RG</td>
<td>0.77</td>
<td>0.84</td>
</tr>
<tr>
<td>WS353</td>
<td>0.60</td>
<td>0.68</td>
</tr>
</tbody>
</table>
SemEval 2012: Sentence Level Semantic Similarity

- BLEU-based semantic similarity metric:
  - Baseline BLEU: using single BLEU hit rate as rating
  - Semantic Similarity (SS) BLEU: modified unigram BLEU that includes semantic similarity of non-matched words

<table>
<thead>
<tr>
<th></th>
<th>par</th>
<th>vid</th>
<th>euro</th>
<th>Mean</th>
<th>Ovrl</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>0.54</td>
<td>0.60</td>
<td>0.39</td>
<td>0.51</td>
<td>0.58</td>
</tr>
<tr>
<td>SS-BLEU WordNet</td>
<td>0.56</td>
<td>0.64</td>
<td>0.41</td>
<td>0.54</td>
<td>0.58</td>
</tr>
<tr>
<td>SS-BLEU $l(i,j)$</td>
<td>0.56</td>
<td>0.63</td>
<td>0.39</td>
<td>0.53</td>
<td>0.59</td>
</tr>
<tr>
<td>SS-BLEU $l_a(i,j)$</td>
<td><strong>0.57</strong></td>
<td><strong>0.64</strong></td>
<td>0.40</td>
<td><strong>0.54</strong></td>
<td>0.58</td>
</tr>
</tbody>
</table>
Contributions

Proposed a language agnostic, unsupervised and scalable algorithm for semantic similarity computation

- No linguistic knowledge required, works from text corpus or using a web query engine
- Shown to perform at least as well as resource-based semantic similarity computation algorithms, e.g., WordNet-based methods
EmotiWord: Affective Lexicon Creation with Application to Interaction and Multimedia Data
Motivation

- Affective text labeling at the core of many multimedia applications, e.g.,
  - Sentiment analysis
  - Spoken dialogue systems
  - Emotion tracking of multimedia content
- **Affective lexicon** is the main resource used to bootstrap affective text labeling
  - Lexica are currently of **limited scope** and **quality**
Goals and Contributions

Our goal: assigning continuous high-quality polarity ratings to any lexical unit

- We present a method of expanding an affective lexicon, using web-based semantic similarity.
- Assumption: semantic similarity implies affective similarity.
- The expanded lexica are accurate and broad in scope, e.g., they can contain proper nouns, multi-word terms.
Our lexicon expansion method

Expansion of [Turney and Littman, ’02]. Assumption: the valence of a word can be expressed as a linear combination of its semantic similarities to a set of seed words and their valence ratings:

\[ \hat{v}(w_j) = a_0 + \sum_{i=1}^{N} a_i v(w_i) d(w_i, w_j), \] (1)

- \( w_j \): the wanted word
- \( w_1...w_N \): seed words
- \( v(w_i) \): valence rating of word \( w_i \)
- \( a_i \): weight assigned to seed \( w_i \)
- \( d(w_i, w_j) \): measure of semantic similarity between words \( w_i \) and \( w_j \)
Given

- an initial lexicon of $K$ words
- a set of $N < K$ seed words

we can use (1) to create a system of $K$ linear equations with $N + 1$ unknown variables:

$$
\begin{bmatrix}
1 & d(w_1, w_1) v(w_1) & \cdots & d(w_1, w_N) v(w_N) \\
\vdots & \vdots & \ddots & \vdots \\
1 & d(w_K, w_1) v(w_1) & \cdots & d(w_K, w_N) v(w_N)
\end{bmatrix}
\begin{bmatrix}
a_0 \\
a_1 \\
\vdots \\
a_N
\end{bmatrix}
= \begin{bmatrix}
1 \\
v(w_1) \\
\vdots \\
v(w_K)
\end{bmatrix}
$$

Solving with Least Mean Squares estimation provides the weights $a_i$. 

Alexandros Potamianos
Dept. of ECE, Technical Univ. of Crete, Chania, Greece
Lexical Cognitive Networks with Application to Semantic Similarity Computation and Affective Text Analysis
Example, $N = 10$ seeds

<table>
<thead>
<tr>
<th>Order</th>
<th>$w_i$</th>
<th>$v(w_i)$</th>
<th>$a_i$</th>
<th>$v(w_i) \times a_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>mutilate</td>
<td>-0.8</td>
<td>0.75</td>
<td>-0.60</td>
</tr>
<tr>
<td>2</td>
<td>intimate</td>
<td>0.65</td>
<td>3.74</td>
<td>2.43</td>
</tr>
<tr>
<td>3</td>
<td>poison</td>
<td>-0.76</td>
<td>5.15</td>
<td>-3.91</td>
</tr>
<tr>
<td>4</td>
<td>bankrupt</td>
<td>-0.75</td>
<td>5.94</td>
<td>-4.46</td>
</tr>
<tr>
<td>5</td>
<td>passion</td>
<td>0.76</td>
<td>4.77</td>
<td>3.63</td>
</tr>
<tr>
<td>6</td>
<td>misery</td>
<td>-0.77</td>
<td>8.05</td>
<td>-6.20</td>
</tr>
<tr>
<td>7</td>
<td>joyful</td>
<td>0.81</td>
<td>6.4</td>
<td>5.18</td>
</tr>
<tr>
<td>8</td>
<td>optimism</td>
<td>0.49</td>
<td>7.14</td>
<td>3.50</td>
</tr>
<tr>
<td>9</td>
<td>loneliness</td>
<td>-0.85</td>
<td>3.08</td>
<td>-2.62</td>
</tr>
<tr>
<td>10</td>
<td>orgasm</td>
<td>0.83</td>
<td>2.16</td>
<td>1.79</td>
</tr>
<tr>
<td>-</td>
<td>$w_0$ (offset)</td>
<td>1</td>
<td>0.28</td>
<td>0.28</td>
</tr>
</tbody>
</table>
Sentence Tagging

Simple combinations of word ratings:

- linear (average)

\[ v_1(s) = \frac{1}{N} \sum_{i=1}^{N} v(w_i) \]

- weighted average

\[ v_2(s) = \frac{1}{N} \frac{1}{\sum_{i=1}^{N} |v(w_i)|} \sum_{i=1}^{N} v(w_i)^2 \cdot \text{sign}(v(w_i)) \]

- max

\[ v_3(s) = \max_{i} (|v(w_i)|) \cdot \text{sign}(v(w_z)), \quad z = \arg \max_{i} (|v(w_i)|) \]
N-gram Affective Models

- Generalize method to $n$-grams

$$v_i(s) = a_0 + a_1 v_i(\text{unigram}) + a_2 v_i(\text{bigram})$$

- Starting from all 1-grams and 2-grams, select terms:
  1. **Backoff**: use overlapping bigrams as default, revert to unigrams based on mutual information-based criterion
  2. **Weighted interpolation**: use all unigrams and bigrams as default, reject bigrams based on criterion

- In both cases unigrams and bigrams are given linear weights, trained using LMS
Evaluation

- **ANEW** Word Polarity Detection Task
  - Affective norms for English words (ANEW) corpus
  - 1,034 English words, continuous valence ratings

- **General Inquirer** Word Polarity Detection
  - General Inquirer words corpus
  - 3,607 English words, binary valence ratings

- **BAWLR** Word Polarity Detection Task
  - Berlin affective word list reloaded (BAWLR) corpus
  - 2,902 German words, continuous valence ratings

- **SemEval 2007** Sentence Polarity Detection
  - SemEval 2007 News Headlines corpus
  - 1,000 English sentences, continuous valence ratings
  - ANEW used for lexicon training
  - 250 sentence development set used for word fusion training
Experimental Procedure

- **Corpus selection**
  - Web corpus (web)
  - Lexically balanced web corpus (14m, 116m)

- **Semantic Distance**
  - Co-occurrence based (G = google)
  - Context-based using web snippets (S)

- All experiments: training on ANEW seed words (cross-validation)
Word Polarity Detection (ANEW)

2-class word classification accuracy (positive vs negative)
Word Polarity Detection (GINQ)

2-class word classification accuracy (positive vs negative)
Word Polarity Detection (BAWLR)

2-class word classification accuracy (positive vs negative)
Sentence Polarity Detection (SemEval 2007)

2-class sentence classification accuracy (positive vs negative), using weighted interpolation

![Graph showing accuracy vs number of seeds]

Alexandros Potamianos
Dept. of ECE, Technical Univ. of Crete, Chania, Greece

Lexical Cognitive Networks with Application to Semantic Similarity Computation and Affective Text Analysis
Sentence Polarity Detection (SemEval 2007)

2-class sentence classification accuracy (positive vs negative),
vs bigram rejection threshold

Alexandros Potamianos
Dept. of ECE, Technical Univ. of Crete, Chania, Greece

Lexical Cognitive Networks with Application to Semantic Similarity Computation and Affective Text Analysis
ChIMP Sentence Frustration/Politeness Detection

- ChIMP Children Utterances corpus
- 15,585 English sentences, Politeness/Frustration/Neutral ratings
- SoA results, binary accuracy P vs 0 / F vs O:
  - 81% / 62.7% [Yildirim et al, ’05]
- 10-fold cross-validation
- ANEW used for training/seeds to create word ratings
- ChiMP words added to ANEW with weight $w$, to adapt to the task
- Similarity metric: Google semantic relatedness
- Only content words taken into account
<table>
<thead>
<tr>
<th>Politeness: Sentence</th>
<th>Fusion scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification Accuracy</td>
<td>avg</td>
</tr>
<tr>
<td>Baseline: P vs O</td>
<td>0.70</td>
</tr>
<tr>
<td>Adapt $w = 1$: P vs O</td>
<td>0.74</td>
</tr>
<tr>
<td>Adapt $w = 2$: P vs O</td>
<td>0.77</td>
</tr>
<tr>
<td>Adapt $w = \infty$: P vs O</td>
<td><strong>0.84</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frustration: Sentence</th>
<th>Fusion scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification Accuracy</td>
<td>avg</td>
</tr>
<tr>
<td>Baseline: F vs O</td>
<td>0.53</td>
</tr>
<tr>
<td>Adapt $w = 1$: F vs O</td>
<td>0.51</td>
</tr>
<tr>
<td>Adapt $w = 2$: F vs O</td>
<td>0.49</td>
</tr>
<tr>
<td>Adapt $w = \infty$: F vs O</td>
<td>0.52</td>
</tr>
</tbody>
</table>

Alexandros Potamianos  
Dept. of ECE, Technical Univ. of Crete, Chania, Greece  
Lexical Cognitive Networks with Application to Semantic Similarity Computation and Affective Text Analysis
Summary of Results

- The word-level ratings are very **accurate** and **robust** across different corpora.
- N-gram sentence-level ratings are **significantly better than the state-of-the-art**, despite the simplistic sentence level fusion model and disregard of syntax/negations.
- **Adaptation** provided good performance on the **politeness** detection task (linear fusion).
- The **baseline model** performed best on the **frustration** detection task (max fusion).

Alexandros Potamianos

Dept. of ECE, Technical Univ. of Crete, Chania, Greece

Lexical Cognitive Networks with Application to Semantic Similarity Computation and Affective Text Analysis
Conclusions

Proposed a high-performing, robust, general-purpose and scalable algorithm for affective lexicon creation

- Investigated linear and non-linear sentence level fusion schemes, showing good but task-dependent performance
- Investigated domain adaptation with good but task-dependent performance (politeness vs frustration detection task)
- Demonstrated that distributional approach can generalize to n-grams
Ongoing Work

- Similarity metrics on **semantic networks**
  - Graph theoretic approaches, e.g., cliques
  - Local and global normalization
- *(Non-)compositional models* **Semantics and Affect:**
  - Additional information, modifiers, functionals: syntax, negations, modifiers
  - **Fusion** of semantic and distributional models
  - Temporal integration of sentence ratings
  - Modeling context and **affective reversal**
- **Cognitive models of semantics and affect**
  - Low dimensional semi-metric semantic spaces