Building Lexical Cognitive Networks for Web Corpora with Application to Lexical 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

[1] E. Iosif and A. Potamianos. 2010. “Unsupervised semantic similarity computation between terms using web documents”. IEEE Transactions on Knowledge and Data Engineering.
[2] N. Malandrakis, A. Potamianos, E. Iosif, S. Narayanan. 2011. “Kernel methods for affective lexicon creation”. Proc. Interspeech.
[3] — . 2011. “EmotiWord: Affective Lexicon Creation with Application to Interaction and Multimedia Data”. Proc. of MUSCLE workshop.
[4] E. Iosif and A. Potamianos. 2012. “Semsim: Resources for normalized semantic similarity computation using lexical networks”. In Proc. LREC.
[5] N. Malandrakis, E. Iosif, A. Potamianos. 2012. “DeepPurple: Estimating Sentence Semantic Similarity using N-gram Regression Models and Web Snippets”. In Proc SemEval (collocated with NAACL-HLT).
[6] E. Iosif and A. Potamianos. 2012. “Similarity computation using semantic networks created from web-harvested data”. Natural Language Engineering (submitted to).

Semantic Similarity Computation

- Compute semantic similarity between words $S(i, j)$
  - Organizing principle of human cognition
  - Building block of machine learning in NLP/semantic web
  - Underlies the relations between words

How Humans do it?

- How is lexical information organized cognitively?
- Do people think with words, i.e., are words the building blocks of human cognition?
- Do you believe in word senses?
- Affective organization of words?

How Humans do it?

- Priming: network-based activation
- Framing: effect of context
- Associative anchoring
- Valence reversal

- Semantic similarity estimation methods:
  - 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

Max. sense sim. assumption: similarity of two closest senses

- fruit
  - Sense1: *"the ripened reproductive body of a seed plant"*
  - Sense2: *"an amount of a product"*
  - Sense3: *"the consequence of some effort or action"*
- tree
  - Sense1: *"a tall perennial woody plant ..."*
  - Sense2: *"a figure that branches from a single root"*
- forest
  - Sense1: *"trees and other plants in a densely wooded area"*
  - Sense2: *"land that is covered with trees and shrubs"*

Semantic Sim. Computation: Attributional Similarity

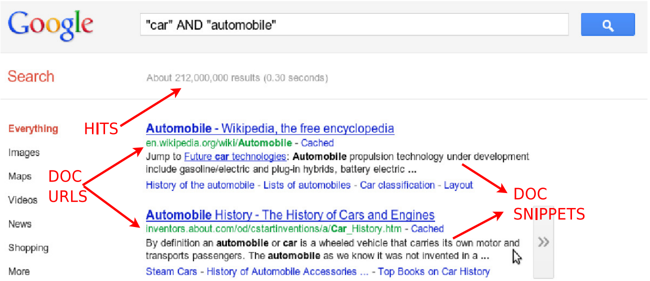
Attributional similarity assumption

- 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)

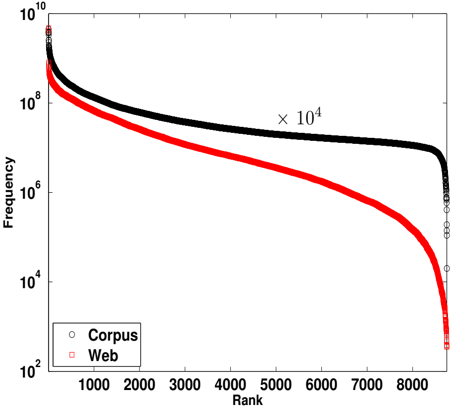
Enter semantic networks

- 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

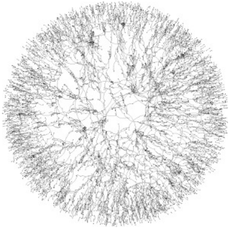
Corpus and Network Creation

- 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

Corpus: Frequency vs. Rank

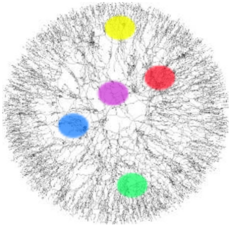


Lexical Network - Semantic Neighborhoods



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

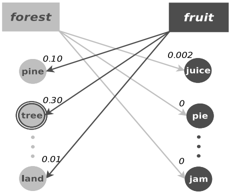
Semantic Neighborhoods: Examples

Word	Neighbors
automobile	**auto**, truck, **vehicle**, **car**, **engine**, bus, . . .
car	truck, **vehicle**, travel, service, **price**, **industry**, . . .
slave	slavery, beggar, **nationalism**, society, **democracy**, **aristocracy**, . . .
journey	**trip**, holiday, **culture**, **travel**, **discovery**, **quest**, . . .

- Synonymy
- Taxonomic: IsA, Meronymy
- Associative
- Broader semantics/pragmatics
- . . .

Neighborhood-based Similarity Metrics: M_n

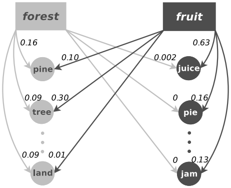
M_n metric: maximum similarity of neighborhoods



- 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

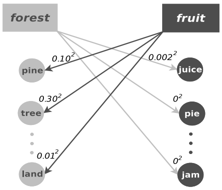
R_n metric: correlation of neighborhood similarities



- 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
  - $r_1((0.16 \ldots 0.09), (0.10 \ldots 0.01)), r_2((0.002 \ldots 0), (0.63 \ldots 0.13))$
- Select max. correlation: $R_n(\text{“forest”}, \text{“fruit”}) = -0.04$

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 $\theta = 2$
- $E_n^{\theta=2}$(*"forest"*, *"fruit"*)= $\sqrt{(0.10^2 + \cdots + 0.01^2) + (0.002^2 + \cdots + 0^2)} = 0.22$

Minimum Error Sem. Similarity: Problem Definition

- Goal: reduce the similarity estimation error
  - Follow max. sense similarity assumption
  - Modify standard metrics
  - Case study: co-occurrence-based metrics
- Consider metric $S_W(w_i, w_j) = \frac{\hat{p}(w_i,w_j)}{\hat{p}(w_i)\hat{p}(w_j)}$
  - $\hat{p}(w_i)$ and $\hat{p}(w_j)$: occur. prob. for words w_i and w_j
  - $\hat{p}(w_i, w_j)$: co-occur. prob. of w_i and w_j
- Problem: error in $S_W(w_i, w_j)$ due to:
  - Estimation of $\hat{p}(w_i, w_j)$
    - w_i and w_j co-occur with close senses?
    - scope (doc, sentence, syntactic rel., ...) of co-occurrence?
  - $\hat{p}(w_i)$, $\hat{p}(w_j)$ estimated across all senses of w_i, w_j

Minimum Error Sem. Similarity: Assumptions

- Set of words $L = \{w_1, w_2, ...w_N\}$
- Set of senses for word w_i: $M_i = \{s_{i1}, s_{i2}, ..., s_{iN_i}\}$
- Set of senses of all words: $M = M_1 \cup M_2 \cup ...M_N$
- Assumption 1
  - All senses lexicalized as single words included in L

$$\forall s_{ij} \in M, \exists w_k \in L : s_{ij} \equiv w_k$$

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

$$S_W(w_i, w_j) \equiv S_S(s_{ik}, s_{jl}), \quad (k, l) = \underset{(p \in M_i, r \in M_j)}{\operatorname{argmax}} S_S(s_{ip}, s_{jr})$$

Minimum Error Sem. Similarity: Assumptions

- Assumption 3
  - [3a] w_i, w_j always co-occur with their two closest senses

$$\forall\{w_i * w_j\} : (w_i \equiv s_{ik}, w_j \equiv s_{jl}) \text{ iff } (k, l) = \underset{(p \in M_i, r \in M_j)}{\text{argmax}} S_S(s_{ip}, s_{jr})$$

  - [3b] As [3a] with extra, small prob. $\epsilon_1 = f(p(w_i)p(w_j))$

$$p(w_i, w_j) \equiv p(s_{ik}, s_{jl}) + \epsilon_1$$

- Assumption 4
  - [4a] Uniform sense distr.: $\forall k : p(s_{ik}) = \frac{p(w_i)}{N_i}$
  - [4b] Power-law sense distr.: $\forall k : p(s_{ik}) = f(p(w_i)^\alpha)$

Evaluation: Word Level Semantic Similarity

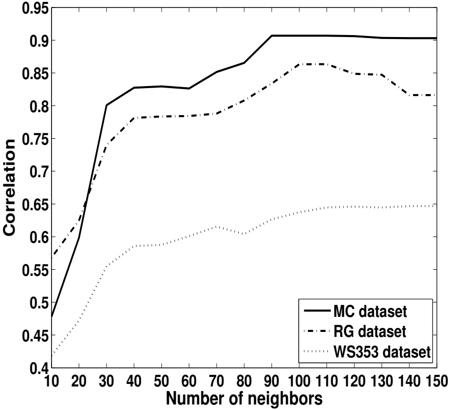
- Task: similarity judgment
  - Noun pairs
- Datasets
  - MC *[Miller and Charles, 1998]*
  - RG *[Rubenstein and Goodenough, 1965]*
  - WS353 *[Finkelstein et al., 2002]*
- Evaluation metric: correlation wrt to human ratings
  - Pearson's correlation coefficient

Performance of net-based similarity metrics

Dataset	Neighbor selection	Similarity computation	Metrics $M_{n=100}$	Metrics $R_{n=100}$	Metrics $E^{\theta=2}_{n=100}$
MC	co-occur.	co-occur.	0.90	0.72	**0.90**
MC	co-occur.	context	**0.91**	0.28	0.46
MC	context	co-occur.	0.52	**0.78**	0.56
MC	context	context	0.51	0.77	0.29
RG	co-occur.	co-occur.	**0.87**	0.67	**0.86**
RG	co-occur.	context	0.86	0.32	0.53
RG	context	co-occur.	0.58	**0.72**	0.61
RG	context	context	0.57	0.69	0.33
WS353	co-occur.	co-occur.	**0.64**	0.50	**0.64**
WS353	co-occur.	context	**0.64**	0.14	0.20
WS353	context	co-occur.	0.47	0.56	0.48
WS353	context	context	0.46	**0.57**	0.11

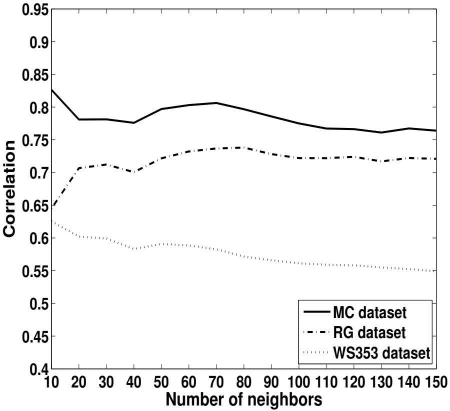
Performance of maximum sim. of neigh. M_n

- Neighbor selection: co-occurrence-based metric
- Similarity computation: context-based metric



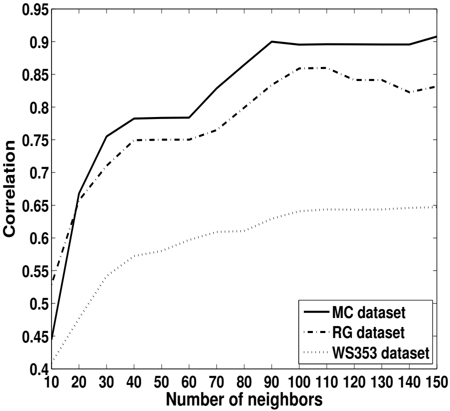
Performance of correlation of neigh. sim. R_n

- Neighbor selection: context-based metric
- Similarity computation: co-occurrence-based metric



Performance of sum of squared neigh. sim. $E_n^{\theta=2}$

- Neighbor selection: co-occurrence-based metric
- Similarity computation: co-occurrence-based metric



Performance of web-based similarity metrics

- For MC dataset

Feature	Description	Correlation
context	AND queries	0.88
context	IND queries	0.55
context	IND queries: network	0.90

- Comparable to structured DSMs, WordNet-based approaches

Performance of min. error sem. sim. (current results)

- 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
- α estimated to max. sense coverage of sem. neigh.
- Task: similarity judgment, correlation wrt to human ratings

Dataset	I	I_α
MC	0.78	**0.89**
RG	0.77	**0.84**
WS353	0.60	**0.68**

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

Correlation performance of 1-gram BLEU scores with semantic similarity metrics (nouns-only)

	par	vid	euro	Mean	Ovrl
BLEU	0.54	0.60	0.39	0.51	0.58
SS-BLEU WordNet	0.56	**0.64**	**0.41**	**0.54**	0.58
SS-BLEU $I(i,j)$	0.56	0.63	0.39	0.53	**0.59**
SS-BLEU $I_a(i,j)$	**0.57**	**0.64**	0.40	**0.54**	0.58

Contributions

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

- No linguistic knowledge required, works from text corpus or from 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 contiunous 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:

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

- 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:

$$\begin{bmatrix} 1 & d(w_1,w_1)v(w_1) & \cdots & d(w_1,w_N)v(w_N) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & d(w_K,w_1)v(w_1) & \cdots & d(w_K,w_N)v(w_N) \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} 1 \\ v(w_1) \\ \vdots \\ v(w_K) \end{bmatrix} \quad (2)$$

Solving with Least Mean Squares estimation provides the weights a_i.

Example, $N = 10$ seeds

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

Sentence Tagging

Simple combinations of word ratings:

- linear (average)

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

- weighted average

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

- max

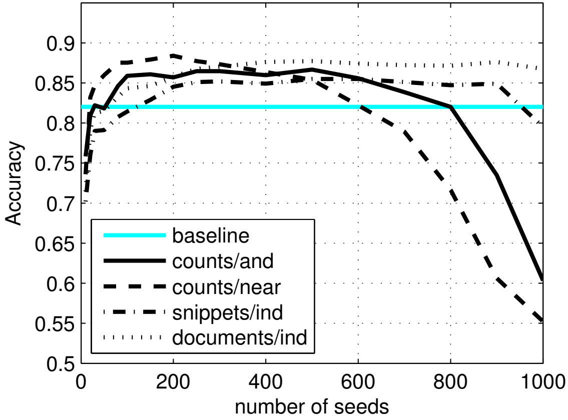
$$v_3(s) = \max_i(|v(w_i)|) \cdot \mathrm{sign}(v(w_z)), \quad z = \arg\max_i(|v(w_i)|)$$

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
- SemEval 2007 Sentence Polarity Detection
  - SemEval 2007 News Headlines corpus
  - 1.000 English sentences, continuous valence ratings
  - ANEW used for training

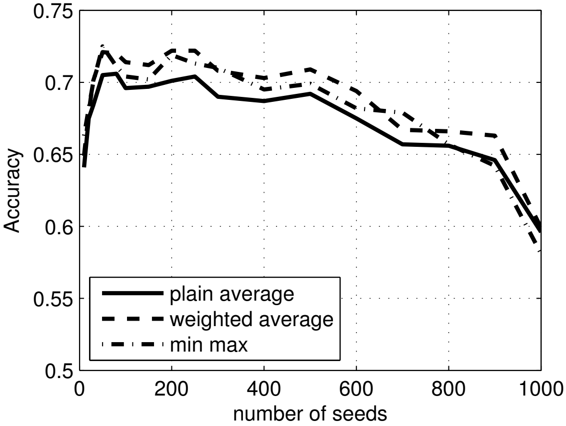
Word Polarity Detection (ANEW)

2-class word classification accuracy (positive vs negative)



Sentence Polarity Detection (SemEval 2007)

2-class sentence classification accuracy (positive vs negative)



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

Politeness: Sentence Classification Accuracy	Fusion scheme		
	avg	w.avg	max
Baseline: P vs O	0.70	0.69	0.54
Adapt $w = 1$: P vs O	0.74	0.70	0.67
Adapt $w = 2$: P vs O	0.77	0.74	0.71
Adapt $w = \infty$: P vs O	**0.84**	0.82	0.75

Frustration: Sentence Classification Accuracy	Fusion scheme		
	avg	w.avg	max
Baseline: F vs O	0.53	0.62	**0.66**
Adapt $w = 1$: F vs O	0.51	0.58	0.57
Adapt $w = 2$: F vs O	0.49	0.53	0.53
Adapt $w = \infty$: F vs O	0.52	0.52	0.52

Summary of Results

- The word-level ratings are very accurate and robust across different corpora
- Sentence-level ratings comparable to 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)

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)

Future Work

- (Non-)compositional Semantics and Affect:
  - Investigate word fusion models
  - Additional information, modifiers, functionals: syntax, negations, modifiers
  - Temporal integration of sentence ratings
  - Multilinguality
- Cognitive models of semantics and affect