A Genetic Algorithm using Semantic Relations for Word Sense Disambiguation

Master’s Project

Michael Hausman
Overview

• Introduction
• Semantic Relations
• Genetic Algorithm
  – Cost Function
  – Mating
  – Mutation
• Results
Introduction

• Word Sense Disambiguation
  – Another way of saying, “which dictionary definition is correct in context”
  – Main problem of the project
• Example: “Time flies like an arrow.”
  – Time (noun #1) -- an instance or single occasion for some event
  – Time (noun #5) -- the continuum of experience in which events pass from the future through the present to the past
  – Fly (noun #1) -- two-winged insects characterized by active flight
  – Fly (verb #1) -- travel through the air; be airborne;
  – Fly (verb #2) -- move quickly or suddenly
  – Etc.
• Objective
  – To demonstrate that a simple algorithm can achieve comparable results to the best algorithms for Word Sense Disambiguation
How to Disambiguate

- Example: *The boy walked down the hillside to the river bank and went for a swim.*
  - Is *bank* “sloping land” or a “financial institution?”
- Humans look at the surrounding words to find the context
- Many words are related
  - *river bank* is a compound word
  - *hillside* and *bank* (sloping land) are both geological features
  - *river* is a body of water and *swim* is a water sport
  - Money is difficult to swim in, except for Scrooge McDuck
- A machine can use some of these relations
Related Work

- Zhang et al. (2008)
  - Use three semantic relations as the cost function in a very simple implementation of a genetic algorithm for nouns
  - Took the idea of semantic relations in a genetic algorithm
- Basile et al. (2007)
  - JIGSAW: Use a different algorithm and/or weights for every part of speech. Nouns and verbs use modified hypernym measures. Adjectives and Adverbs use Adapted Lesk.
  - Took the idea of using different measures based on part of speech
- Patwardhan et al. (2007)
  - UMND1: Only use semantic relations from WordNet for the Lexical Sample task in the SemEval 2007 competition*
  - Proof that others are trying semantic relations as an unsupervised approach
- Decadt et al. (2004)
  - GAMBL: Uses a genetic algorithm and several corpuses to optimize two TIMBL (memory based learning) classifiers
  - The extra complication seems excessive

*Lexical sample task is word sense disambiguation for a single word in a sentence while this project focuses on every word
Semantic Relations

• Semantic Relations use some aspect of a word to define how closely two objects are related

• Early papers focus on one semantic relation
  – Somewhat good results

• Later papers focus on two or three relations
  – Better results over early papers

• Therefore, this project uses several semantic relations
  – Frequency, Hypernyms, Coordinate Sisters, Domain, Synonyms, Antonyms
Tools/Resources

• Part of Speech Tagger
  – From Tokyo University
  – 97.10% accuracy on WSJ corpus

• WordNet
  – Lexical database
  – Machine readable dictionary
  – Contains all the semantic relations in this project

• SemCor
  – 20,000+ tagged words across 352 files
  – Tagged with WordNet senses
Semantic Relations: Frequency

- WordNet orders senses (definitions) by how often it appears in the corpus used to make WordNet
- More common senses are more likely to be correct

\[
Freq(w) = \begin{cases} 
\frac{SenseCnt(w)}{TotalCnt(w)}, & \text{TotalCnt}(w) > 0 \\
\frac{SenseTotal(w) - Sense(w) - 1}{SenseTotal(w)}, & \text{Otherwise}
\end{cases}
\]

- SenseCnt(w): The number of times this was referenced in WordNet corpus
- TotalCnt(w): The total number of references for this word in WordNet corpus
- SenseTotal(w): The total number of senses for this word
- Sense(w): The sense number of the word currently in use

**Frequency**

<table>
<thead>
<tr>
<th>WordNet senses of the noun bank</th>
<th>883/1130 = 0.78</th>
<th>(883) depository financial institution, bank, banking concern, banking company -- (a financial institution that accepts deposits and channels the money into lending activities)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>76/1130 = 0.07</td>
<td>(99) bank -- (sloping land (especially the slope beside a body of water))</td>
</tr>
<tr>
<td></td>
<td>-/1130 = &lt;0.08</td>
<td>...</td>
</tr>
</tbody>
</table>
Semantic Relations: Hypernyms

• A more generic way of saying a word
• Many hypernyms in a row creates a tree
  – More specific down the tree
  – Length of path between words is the similarity of the subjects
• Every definition has a different hypernym tree
  – Bank: financial institution vs. bank: ground next to a river

\[ Hyp(w_1, w_2) = \frac{2 \times D}{A + B + 2 \times D} \]

LSO: most specific common hypernym
D: length from the root to LSO
A: length from \( w_1 \) to LSO
B: length from \( w_2 \) to LSO
\( Hyp(\text{boy}, \text{girl}) \approx 0.714 \)
Semantic Relations: Others

• All four either are in the same group (returns 1), or are not (returns 0)

1. Coordinate Sisters
   – Two words that have the same hypernym are coordinate sisters

2. Domain
   – The word or collection a word belongs to
   – Example: words in the domain *computer science*
     • Buffer, drive, cache, program, software

3. Synonym
   – Two words that can be interchanged in a given context are synonyms

4. Antonym
   – Two words that are opposites of each other in a given context are antonyms
Genetic Algorithms: Overview

• Based off of Darwin’s theory of Evolution
• Use a subset of total number of solutions to evolve a “good” answer over time

1. Start with a set of solutions (1st Generation)
2. Take original “parent” solutions and combine them with each other to create a new set of “child” solutions (Mating)
3. Introduce some random changes in case solutions are “stuck” or are all the same (Mutation)
4. Somehow measure the solutions to evaluate the best solution (Cost Function)
5. Repeat starting with Step 2
Cost Function

• Determines which solution is “better” when comparing solutions
• Most important part of genetic algorithm
• Needs to be able to determine the correct answer
  – Otherwise the genetic algorithm converges on the wrong solution
• Higher value means a better solution
• This project has three main calculations
  – Semantic Relation Ratio
  – Sense Distribution
  – Semantic Relation Distribution
Sense Combinations from SemCor

Frequency of Noun-Noun

Correct Area

Separate the Parts of Speech

Most Frequent Senses

Percentage of Words Correct

Set of Sense Combinations

Least Frequent Senses

Average Frequency Value for All Words

Correct (%)

Frequency

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
Cost Function: Semantic Relation Ratio

1. Take several sense combinations for one SemCor file
2. Record the min, max, and ratio of the correct answer for each Semantic Relation
3. Find the average and standard deviation of the correct solution ratio from multiple SemCor files
4. Use four lines to transform the result for a Semantic Relation

![Proportional Range Example](image)

- (Min, 0)
- (Max, 0)
- (Avg, stdDev^2)
- (Avg – 2*stdDev, 0.9*stdDev^2)
- (Avg + 2*stdDev, 0.9*stdDev^2)
Cost Function: Sense Distribution

1. Take several SemCor files and find the average percentage of each sense

2. Find the error using the following equation. It basically adds the error for each sense and subtracts it from one.

\[
Weight(s) = 1 - \sum_{\text{Sense}} \frac{\text{Abs}(\text{TotalCnt}(\text{sense}) - \text{SenseCnt}(\text{sense}))}{\text{TotalCnt}(\text{sense})}
\]

S: The current solution
Sense: The sense number
SenseCnt: Find the current number of the given sense
TotalCnt: The expected number of the given sense
Abs: Absolute value
Cost Function: Semantic Relation Distribution

1. Take several SemCor files and find the average semantic relation value for each sense

2. Find the error using the following equation. It basically adds the error for each sense and subtracts it from one.

\[
\text{Weight}(\text{SemRel}) = 1 - \sum_{\text{Sense}} \frac{\text{Abs}(\text{SemOpt}(\text{sense}) - \text{SemVal}(\text{sense}))}{\text{SemOpt}(\text{sense})}
\]

SemRel: The current semantic relation
Sense: The sense number
SemVal: The average semantic relation value for this sense
SemOpt: The optimal semantic relation value for this sense
Abs: Absolute value
Cost Function: Final Equation

This equation

1. Adds the Semantic Relation contributions for the word pairs in a sliding window
2. Transforms the average with the Semantic Relation Ratio
3. Weights it with the Semantic Relation Distribution
4. Weights the total semantic relation sum with the Sense Distribution

\[
Cost(s) = \text{Weight(Sense)} \ast \sum_{\text{SemRel}} \left( \text{Weight(SemRel)} \right)
\]

\[
\ast \text{Transform} \left( \sum_{\text{Word}_1} \sum_{\text{Word}_2} \text{SemRelEqn} \left( \text{SemRel, Word}_1, \text{Word}_2 \right) \right)
\]

S: The current solution
Word\(_1\): Every word in the system
Word\(_2\): The surrounding 10 words before and after word\(_1\)
SemRel: Every semantic relation
SemRelEqn: Use the words in the correct semantic relation equation
Transform: Find the average value then transform with the proportional equation
Dominant Gene Genetic Algorithm

- Apply the cost function at a gene level
  - Apply to every word
  - Words with high cost are most likely to be correct

- Mating functions focus on high cost words (dominant genes)
  - Two variations in this project

- Mutation functions focus on low cost words (recessive genes)
  - Four variations in this project
Mating Function: Mate Top Third

1. Find Dominant Genes
   Parent 1 Words:  a1  a2  a3  a4  a5  a6  ...  
   Gene Cost:      11  12  3  24  15  4  ...  
   Select the upper 1/3
   Dominant Words: a4, a5, etc

   Parent 2 Words:  b1  b2  b3  b4  b5  b6  ...  
   Gene Cost:      25  15  5  10  8  14  ...  
   Select the upper 2/3
   Dom. Genes: b1, b2, b4, b6, etc

2. Place Dominant Genes Based on First Parent
   Child:  *  *  *  a4  a5  *  ...  

3. Fill in Blanks from Second Parent
   Child:  b1  b2  *  a4  a5  b6  ...  

4. Fill in any Remaining Blanks from First Parent
   Child:  b1  b2  a3  a4  a5  b6  ...
# Mating Function: Mate Middle Third

1. **Find Dominant Genes**
   - Parent 1 Words: \( a_1, a_2, a_3, a_4, a_5, a_6, \ldots \)  
   - Gene Cost: \( 11, 12, 3, 24, 15, 4, \ldots \)  
   - Select the middle 1/3  
   - Dominant Words: \( a_1, a_2, \ldots \)

   - Parent 2 Words: \( b_1, b_2, b_3, b_4, b_5, b_6, \ldots \)  
   - Gene Cost: \( 25, 15, 5, 10, 8, 14, \ldots \)  
   - Select the upper 2/3  
   - Dom. Genes: \( b_1, b_2, b_4, b_6, \ldots \)

2. **Place Dominant Genes Based on First Parent**
   - Child: \( a_1, a_2, * *, * *, * *, \ldots \)

3. **Fill in Blanks from Second Parent**
   - Child: \( a_1, a_2, * *, b_4, * *, b_6, \ldots \)

4. **Fill in any Remaining Blanks from First Parent**
   - Child: \( a_1, a_2, a_3, b_4, a_5, b_6, \ldots \)
Mutation Function: Random Mutation

1. Randomly pick a percentage between 0% and 20%
2. Randomly pick that percentage of words from the solution
3. For each of those words, randomly pick one of the available senses
Mutation Function: Semantic Relation Score Mutation

1. Start with the given semantic relation average
2. Find the percentage this average is off from the optimal semantic relation score
3. Randomly pick the comparison percentage of words
4. If semantic relation cost needs to be increased
   a. If the semantic relation is frequency, then randomly pick a sense lower than the current sense. Otherwise use step b.
   b. Look at each sense starting at the first sense. Stop and select that sense when the sense increases the semantic relation cost.
5. If the semantic relation cost needs to be decreased
   a. If the semantic relation is frequency, then randomly pick a sense higher than the current sense. Otherwise use step b.
   b. Look at each sense starting at the first sense. Stop and select that sense when the sense decreases the semantic relation cost.
Mutation Function: Sense Distribution

1. Find the number of words extra or missing for each sense compared to the optimal sense distributions

2. Find the genes that have the lowest gene cost

3. Look at each sense distribution
   a. If the current sense distribution has extra words, move the lowest cost genes to a sense distribution needing words
Mutation Function: Semantic Relation Distribution

1. Randomly pick a semantic relation

2. Compare each average semantic relation value for each sense to the optimal semantic relation value for that sense. This should result in the number of words that need to change for each sense.

3. Find the genes that have the lowest gene cost

4. Look at each sense distribution
   a. If the current sense distribution has extra words, move the lowest cost genes to a sense distribution needing words
Measuring the Results

- Typically three different measurements to measure “accuracy” for word sense disambiguation

\[
\text{Coverage} = \frac{\text{WordsAnswered}}{\text{TotalWords}}
\]

\[
\text{Recall} = \frac{\text{WordsCorrect}}{\text{TotalWords}}
\]

\[
\text{Precision} = \frac{\text{WordsCorrect}}{\text{WordsAnswered}}
\]
Comparison to Michael Billot

- Colleague at University of Colorado at Colorado Springs
  - Uses a Page Rank algorithm for verbs only
  - Tested using a walkthrough for the game Zelda: Ocarina of Time
  - Evaluated by hand

- Don’t have the walkthrough or senses used
  - “Estimate” by comparing verb SemCor results to Billot’s estimated results
  - Compare to the page rank paper Billot was referencing (2004)

<table>
<thead>
<tr>
<th>Comparison</th>
<th>Results</th>
<th>Hausman</th>
</tr>
</thead>
<tbody>
<tr>
<td>Billot (Verbs only)</td>
<td>100%</td>
<td>46.4%</td>
</tr>
<tr>
<td>Page Rank (SemCor)</td>
<td>100%</td>
<td>45.1%</td>
</tr>
<tr>
<td>Page Rank (SenseEval 2)</td>
<td>100%</td>
<td>50.8%</td>
</tr>
<tr>
<td>Page Rank with Frequency (SemCor)</td>
<td>100%</td>
<td>70.1%</td>
</tr>
<tr>
<td>Page Rank with Frequency (SenseEval 2)</td>
<td>100%</td>
<td>63.3%</td>
</tr>
</tbody>
</table>
Comparison to Another Genetic Algorithm

- Zhang et al. (2008) also use Semantic Relations in a Genetic Algorithm
  - Only focus on Nouns
  - Use SemCor Results
- This project:
  - Solves for every part of speech
  - Is faster (sliding window vs. paragraph)

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Zhang</th>
<th>Hausman (Nouns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Coverage</td>
<td>100%</td>
<td>89.25%</td>
</tr>
<tr>
<td>Avg. Recall</td>
<td>71.96%</td>
<td>55.45%</td>
</tr>
<tr>
<td>Avg. Precision</td>
<td>71.96%</td>
<td>62.13%</td>
</tr>
<tr>
<td># Files &gt;70%</td>
<td>51</td>
<td>9</td>
</tr>
</tbody>
</table>
Comparison to SemEval

- Well known Language Processing competitions
- Occurs approximately every 3 years
- Every competition has a Word Sense Disambiguation task
- Results in the Similar System column
  - Primarily rely on Semantic Relations from WordNet
  - Use SemCor statistics for any weighting

<table>
<thead>
<tr>
<th>Competition</th>
<th>Top Place</th>
<th>Similar System</th>
<th>Hausman</th>
</tr>
</thead>
<tbody>
<tr>
<td>SemEval 2</td>
<td>100%</td>
<td>69.8%</td>
<td>100%</td>
</tr>
<tr>
<td>SemEval 3</td>
<td>100%</td>
<td>65.1%</td>
<td>97.2%</td>
</tr>
<tr>
<td>SemEval 2007</td>
<td>100%</td>
<td>82.5%</td>
<td>100%</td>
</tr>
</tbody>
</table>
Top SemEval Competitors

• Typically use:
  – Keep track of information on the sense level
  – Typically use semantic relations, grammatical information, and anything else they can think of
  – Use multiple training texts
  – Use complicated algorithms
    • SenseEval 3: GAMBL uses a genetic algorithm and several corpuses to optimize two TIMBL classifiers (memory based learning)
    • SenseEval 2007: NUS-PT uses SVMs on several resources
    • SenseEval 2007 (2nd Place): NUS-ML uses a three-level hierarchical bayesian model
  • This project is MUCH simpler
    – Only uses semantic relations between words
    – Single training text
  • Wanted to demonstrate that a simple algorithm can achieve comparable results to the best algorithms for word sense disambiguation
Questions?
References


References


References


