Measuring Semantic Similarity: Representations and Methods

A Dissertation Presented for the Doctor of Philosophy Degree

Mihai C. Lintean

Department of Computer Science
University of Memphis

Under the supervision of Dr. Vasile Rus
Committee Members: Dr. Arthur Graesser, Dr. King-Ip Lin, Dr. Vinhthuy Phan

June 20, 2011

The Goal

Addressing the challenging task of automatically assessing the semantic similarity of texts

The Problem

Text A: York had no problem with MTA's insisting the decision to shift funds had been within its legal rights.

Text B: York had no problem with MTA's saying the decision to shift funds was within its powers.

Paraphrasing - a clear case of semantic similarity

The Problem

Text A:	York had no problem with MTA's insisting the decision to shift funds had been within its legal rights.
Text B:	York had no problem with MTA's saying the decision to shift funds was within its powers.

Paraphrasing - a clear case of semantic similarity

Text A:	About 1,417 schools statewide receive Title I money.
Text B:	That applies only to schools that get federal Title I money.

A clear case when two texts are NOT semantically similar

The Importance of Assessing Semantic Similarity

Applications

- Question Answering Systems
  - compare the input question to a list of known questions

The Importance of Assessing Semantic Similarity

Applications

- Question Answering Systems
  - compare the input question to a list of known questions
- Dialogue-Based Tutoring Systems
  - compare student's answer to a list of known answers

The Importance of Assessing Semantic Similarity

Applications

- Question Answering Systems
  - compare the input question to a list of known questions
- Dialogue-Based Tutoring Systems
  - compare student's answer to a list of known answers
- Text-based Clustering and Classification
  - gather news articles about same story, event or person
  - cluster and classify retrieved documents by their topics

Contributions

- Investigate the role of linguistic information in assessing the semantic similarity of texts
- Propose a Semantic Representation to encode the meaning of natural language texts into structured computational representations
- Design, implement and test a variety of Methods on top of the semantic representation to automatically assess semantic similarity of texts
- Develop a general Framework for assessing the semantic similarity of texts

Outline

Outline

The Problem Reviewed

Text A:	York had no problem with MTA's insisting the decision to shift funds had been within its legal rights.
Text B:	York had no problem with MTA's saying the decision to shift funds was within its powers.

Qualitative Judgement - Paraphrase

Text A:	About 1,417 schools statewide receive Title I money.
Text B:	That applies only to schools that get federal Title I money.

Qualitative Judgement - NOT Paraphrase

The Problem Reviewed

Text A:	York had no problem with MTA's insisting the decision to shift funds had been within its legal rights.
Text B:	York had no problem with MTA's saying the decision to shift funds was within its powers.

Quantitative Judgement - are similar to a degree of 0.9 (on a normalized scale)

Text A:	About 1,417 schools statewide receive Title I money.
Text B:	That applies only to schools that get federal Title I money.

Quantitative Judgement - are similar to a degree of 0.4 (on a normalized scale)

Quantitative Judgement $\Longrightarrow$ Qualitative Judgement

Why Quantitative Analysis

Text A:	Ricky Clemons 's brief, troubled Missouri basketball career is over.
Text B:	Missouri kicked Ricky Clemons off its team, ending his troubled career there.

Why Quantitative Analysis

Text A: Ricky Clemons 's brief, troubled Missouri basketball career is over.

Text B: Missouri kicked Ricky Clemons off its team, ending his troubled career there.

Why Quantitative Analysis

Text A:	Ricky Clemons 's brief, troubled Missouri basketball career is over.
Text B:	Missouri kicked Ricky Clemons off its team, ending his troubled career there.

A *paraphrase* example from the **Microsoft Research Paraphrase (MSR)** Corpus

Symmetric Relation

Why Quantitative Analysis

Text A:	Ricky Clemons 's brief, troubled Missouri basketball career is over.
Text B:	Missouri kicked Ricky Clemons off its team, ending his troubled career there.

A *paraphrase* example from the **Microsoft Research Paraphrase (MSR)** Corpus

Symmetric Relation

Text A:	There are also tanneries, sawmills, textile mills, food-processing plants, breweries, and a film industry in the city.
Text B:	Movies are also made in the city

Why Quantitative Analysis

Text A:	Ricky Clemons 's brief, troubled Missouri basketball career is over.
Text B:	Missouri kicked Ricky Clemons off its team, ending his troubled career there.

A *paraphrase* example from the **Microsoft Research Paraphrase (MSR)** Corpus

Symmetric Relation

Text A:	There are also tanneries, sawmills, textile mills, food-processing plants, breweries, and a film industry in the city.
Text B:	Movies are also made in the city

Why Quantitative Analysis

Text A:	Ricky Clemons 's brief, troubled Missouri basketball career is over.
Text B:	Missouri kicked Ricky Clemons off its team, ending his troubled career there.

A *paraphrase* example from the **Microsoft Research Paraphrase (MSR)** Corpus

Symmetric Relation

Text A:	There are also tanneries, sawmills, textile mills, food-processing plants, breweries, and a film industry in the city.
Text B:	Movies are also made in the city

An *entailment* example from the **Recognizing Textual Entailment (RTE)** Corpus

Asymmetric Relation

Challenges (which we address)

Example #1

Text A:	York had no problem with MTA's insisting the decision to shift funds had been within its legal rights.
Text B:	York had no problem with MTA's saying the decision to shift funds was within its powers.

A *paraphrase* example from the **Microsoft Research (MSR) Paraphrase** Corpus

Challenges (which we address)

Example #1

Text A:	York had no problem with MTA's insisting the decision to shift funds had been within its legal rights.
Text B:	York had no problem with MTA's saying the decision to shift funds was within its powers.

A *paraphrase* example from the **Microsoft Research (MSR) Paraphrase** Corpus

Word-to-word Semantics

Challenges (which we address)

Example #1

Text A:	York had no problem with MTA's insisting the decision to shift funds had been within its legal rights.
Text B:	York had no problem with MTA's saying the decision to shift funds was within its powers.

A *paraphrase* example from the **Microsoft Research (MSR) Paraphrase** Corpus

Word-to-word Semantics

Example #2

Text A:	Besançon is the capital of France's watch and clock-making industry and of high precision engineering.
Text B:	Besançon is the capital of France.

An *non-entailment* example from the **Recognizing Textual Entailment (RTE-1)** Corpus

Challenges (which we address)

Example #1

Text A:	York had no problem with MTA's insisting the decision to shift funds had been within its legal rights.
Text B:	York had no problem with MTA's saying the decision to shift funds was within its powers.

A *paraphrase* example from the **Microsoft Research (MSR) Paraphrase** Corpus

Word-to-word Semantics

Example #2

Text A:	Besançon is the capital of France's watch and clock-making industry and of high precision engineering.
Text B:	Besançon is the capital of France.

An *non-entailment* example from the **Recognizing Textual Entailment (RTE-1)** Corpus

Syntactic Relations between words in a sentence

Challenges (which we do not address)

Example #3

Text A:	That information was first reported in today's edition of the New York Times.
Text B:	The information was first printed yesterday in the New York Times.

Challenges (which we do not address)

Example #3

Text A:	That information was first reported in today's edition of the New York Times.
Text B:	The information was first printed yesterday in the New York Times.

Challenges (which we do not address)

Example #3

Text A:	That information was first reported in today's edition of the New York Times.
Text B:	The information was first printed yesterday in the New York Times.

Need knowledge on: 1) Time 2) Printing business

Challenges (which we do not address)

Example #3

Text A:	That information was first reported in today's edition of the New York Times.
Text B:	The information was first printed yesterday in the New York Times.

Need knowledge on: 1) Time 2) Printing business

Example #4

Text A:	John bought 3 apples and 2 pears.
Text B:	John bought 5 fruits.

Need to know how to Add two integers (Mathematics)

Assessing Semantic Similarity between Texts

- The approach is based on the **Principle of Compositionality**
  - the meaning of a text is determined by the meaning of its constituents and the rules used to combine them

$$\left.\begin{array}{l}\textit{words}\\ \textit{numbers}\\ \textit{punctuation}\end{array}\right\}\ \text{lexical tokens}$$

- How do we compare two words?

Semantic Similarity between Words

- *dog* $\Longleftrightarrow$ *mutt*

Semantic Similarity between Words

- *dog* $\Longleftrightarrow$ *mutt* $\Longleftrightarrow$ *animal*

Semantic Similarity between Words

- *dog* $\Longleftrightarrow$ *mutt* $\Longleftrightarrow$ *animal*

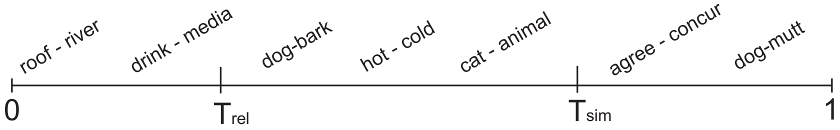
- *dog* $\Longleftrightarrow$ *bark* *apple* $\Longleftrightarrow$ *pie*

Semantic Similarity between Words

- *dog* $\Longleftrightarrow$ *mutt* $\Longleftrightarrow$ *animal*
- *dog* $\Longleftrightarrow$ *bark* *apple* $\Longleftrightarrow$ *pie*
- *hot* $\Longleftrightarrow$ *cold* *agree* $\Longleftrightarrow$ *disagree*

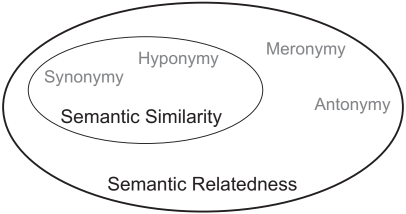
Semantic Similarity between Words

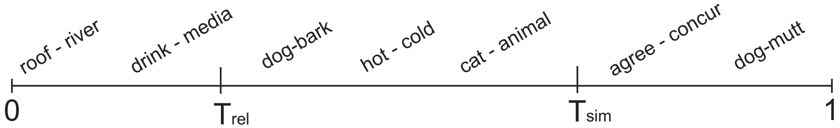
- *dog* $\Longleftrightarrow$ *mutt* $\Longleftrightarrow$ *animal*
- *dog* $\Longleftrightarrow$ *bark* *apple* $\Longleftrightarrow$ *pie*
- *hot* $\Longleftrightarrow$ *cold* *agree* $\Longleftrightarrow$ *disagree*



Semantic Similarity between Words

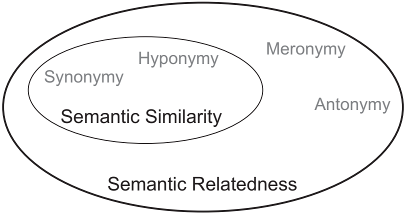
- *dog* $\Longleftrightarrow$ *mutt* $\Longleftrightarrow$ *animal*
- *dog* $\Longleftrightarrow$ *bark* *apple* $\Longleftrightarrow$ *pie*
- *hot* $\Longleftrightarrow$ *cold* *agree* $\Longleftrightarrow$ *disagree*

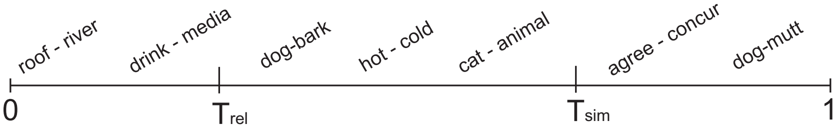




Semantic Similarity between Words

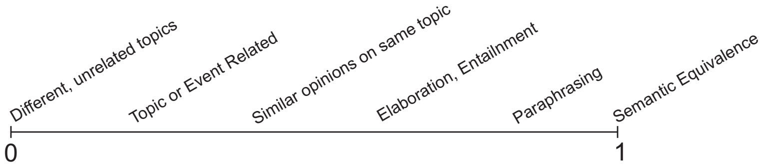
- *dog* $\Longleftrightarrow$ *mutt* $\Longleftrightarrow$ *animal*
- *dog* $\Longleftrightarrow$ *bark* *apple* $\Longleftrightarrow$ *pie*
- *hot* $\Longleftrightarrow$ *cold* *agree* $\Longleftrightarrow$ *disagree*



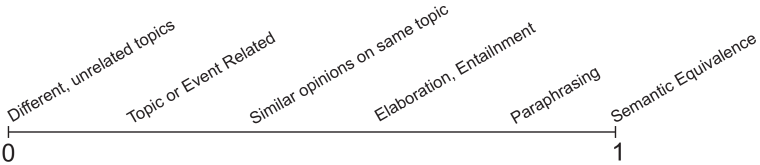


Semantic Similarity versus Semantic Agreement

Semantic Similarity between Sentences



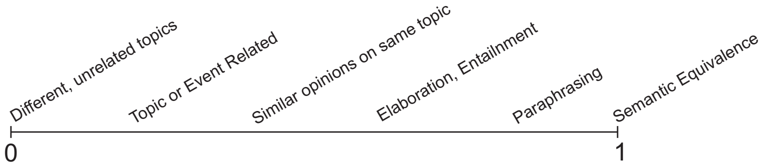
Semantic Similarity between Sentences



Text A:	The Dow finished the volatile day with a modest gain.
Text B:	US stocks rose in volatile trading, thanks only to technical factors.

Same topic...

Semantic Similarity between Sentences



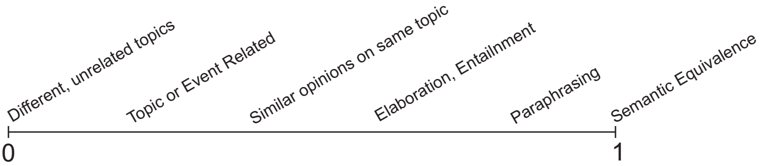
Text A:	The Dow finished the volatile day with a modest gain.
Text B:	US stocks rose in volatile trading, thanks only to technical factors.

Same topic...

Text A:	It is now time to bring our combat troops home from Afghanistan.
Text B:	NATO's secretary general argued against a retreat from Afghanistan.

...but different opinions

Semantic Similarity between Sentences



Text A:	The Dow finished the volatile day with a modest gain.
Text B:	US stocks rose in volatile trading, thanks only to technical factors.

Same topic...

Text A:	It is now time to bring our combat troops home from Afghanistan.
Text B:	NATO's secretary general argued against a retreat from Afghanistan.

...but different opinions

Words $\Longrightarrow$ *Sentences* $\Longrightarrow$ *Paragraphs* $\Longrightarrow$ *Documents*

Previous Work

- Most Common Datasets
  - Recognizing Textual Entailment (RTE) Corpora (PASCAL, TAC)
  - The Microsoft Research (MSR) Paraphrase Corpus (Dolan 04)

Previous Work

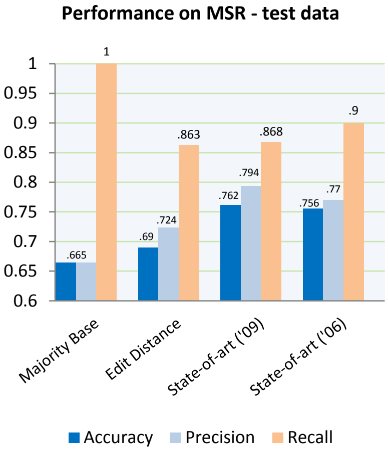
- Most Common Datasets
  - Recognizing Textual Entailment (RTE) Corpora (PASCAL, TAC)
  - The Microsoft Research (MSR) Paraphrase Corpus (Dolan 04)
- A variety of methods
  - word-to-word semantics (Corley & Mihalcea, 05)
  - canonicalized texts (Zhang & Patrick, 05)
  - syntactic dependencies (Lintean & Rus 09, Malakasiotis 09)
  - quasi-synchronous grammars (Das & Smith, 2009)
  - machine translation evaluation metrics (Finch et.al 05, Wan et.al 09)

Previous Work

- Most Common Datasets
  - Recognizing Textual Entailment (RTE) Corpora (PASCAL, TAC)
  - The Microsoft Research (MSR) Paraphrase Corpus (Dolan 04)
- A variety of methods
  - word-to-word semantics (Corley & Mihalcea, 05)
  - canonicalized texts (Zhang & Patrick, 05)
  - syntactic dependencies (Lintean & Rus 09, Malakasiotis 09)
  - quasi-synchronous grammars (Das & Smith, 2009)
  - machine translation evaluation metrics (Finch et.al 05, Wan et.al 09)
- Process outline
  - map the problem into a feature space
  - learn and classify (SVMs, decision trees, logistic regression)

Our Dataset - The MSR Paraphrase Corpus

- identify sentential paraphrases
- 5801 instance pairs
  70% training (.67 T)
  30% testing (.66 T)
- average sentence length:
  17 words
- a challenging dataset
  inconsistent labeling
  83% inter-rater agreement

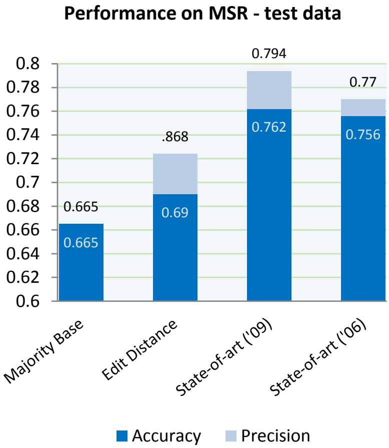


The ULPC Corpus (2000 #instances)

The RTE Corpus (4657 #instances)

Our Dataset - The MSR Paraphrase Corpus

- identify sentential paraphrases
- 5801 instance pairs
  70% training (.67 T)
  30% testing (.66 T)
- average sentence length:
  17 words
- a challenging dataset
  inconsistent labeling
  83% inter-rater agreement



The ULPC Corpus (2000 #instances)

The RTE Corpus (4657 #instances)

A Framework to Measure Semantic Similarity

Our Goal: to offer a fully automated and robust process

- Step1: Semantic Mapping
  - covert the input into semantic representations
  - retain the Lexical, Syntax and Semantics

A Framework to Measure Semantic Similarity

Our Goal: to offer a fully automated and robust process

- Step1: Semantic Mapping
  - covert the input into semantic representations
  - retain the Lexical, Syntax and Semantics
- Step 2: Compare
  - compare the representation
  - extract features that quantify the semantic similarity

A Framework to Measure Semantic Similarity

Our Goal: to offer a fully automated and robust process

- Step1: Semantic Mapping
  - covert the input into semantic representations
  - retain the Lexical, Syntax and Semantics
- Step 2: Compare
  - compare the representation
  - extract features that quantify the semantic similarity
- Step 3: Learn and classify
  - learn from the features
  - assess qualitatively the semantic similarity

A Shallow Representation of Meaning

SR: (*Word*, *Lemma*, *POS*, *Specificity*, WN-SENSE|LSA-Vector,

$$(< - : dep_{type} : dep_{mod} > | < dep_{head} : dep_{type} : - >)+)+$$

Peter went to Seattle last Thursday.

[(Word=*Peter*, lemma=*peter*, POS=NNP, WN_{SENSE}=1,Deps=(*went*:nsubj:-)),
(*went*, *go*, VBP, 1, (-:nsubj:*peter*; -:prep_to:*seattle*; -:tmod:*thursday*)),
(*to*, *to*, N, 1, ()),
(*Seattle*, *seattle*, NNP, 1, (*went*:prep_to:-)),
(*last*, *last*, JJ, 1, (*thursday*:amod:-)),
(*Thursday*, *thursday*, NNP, 1, (*went*:tmod:-; -:amod:*last*))]
(., ., PERIOD, 1, ())]

- Easy extraction of data
- Human friendly
- Encode all lexical, syntactic and semantic facts of the input

A Shallow Representation of Meaning

SR: (*Word*, *Lemma*, *POS*, *Specificity*, WN-SENSE|LSA-Vector,
$(< - : dep_{type} : dep_{mod} > | < dep_{head} : dep_{type} : - >)+)+$

Preprocessing the Input

- Tokenize text $\rightsquigarrow$ lexical tokens (words)

A Shallow Representation of Meaning

SR: (*Word*, *Lemma*, *POS*, *Specificity*, WN-SENSE|LSA-Vector,
$(< - : dep_{type} : dep_{mod} > | < dep_{head} : dep_{type} : - >)+)+$

Preprocessing the Input

- Tokenize text $\rightsquigarrow$ lexical tokens (words)
- Lematize tokens $\rightsquigarrow$ lemmas

A Shallow Representation of Meaning

SR: (*Word*, *Lemma*, *POS*, *Specificity*, WN-SENSE|LSA-Vector,
$(< - : dep_{type} : dep_{mod} > | < dep_{head} : dep_{type} : - >)+)+$

Preprocessing the Input

- Tokenize text $\leadsto$ lexical tokens (words)
- Lematize tokens $\leadsto$ lemmas
- Part-of-Speech Tagging $\leadsto$ POSs

A Shallow Representation of Meaning

SR: (*Word*, *Lemma*, *POS*, *Specificity*, WN-SENSE|LSA-Vector,
$(< - : dep_{type} : dep_{mod} > | < dep_{head} : dep_{type} : - >)+)+$

Preprocessing the Input

- Tokenize text $\rightsquigarrow$ lexical tokens (words)
- Lematize tokens $\rightsquigarrow$ lemmas
- Part-of-Speech Tagging $\rightsquigarrow$ POSs
- Extract Word Specificity from precalculated Indices (IDF, Entropy)

A Shallow Representation of Meaning

SR: (*Word*, *Lemma*, *POS*, *Specificity*, WN-SENSE|LSA-Vector,

$$(< - : dep_{type} : dep_{mod} > | < dep_{head} : dep_{type} : - >)+)+$$

Preprocessing the Input

- Tokenize text $\rightsquigarrow$ lexical tokens (words)
- Lematize tokens $\rightsquigarrow$ lemmas
- Part-of-Speech Tagging $\rightsquigarrow$ POSs
- Extract Word Specificity from precalculated Indices (IDF, Entropy)
- Compute the Meaning of Words $\rightsquigarrow$ WordNet Sense — LSA Vector

A Shallow Representation of Meaning

SR: (*Word*, *Lemma*, *POS*, *Specificity*, WN-SENSE|LSA-Vector,

$$(< - : dep_{type} : dep_{mod} > | < dep_{head} : dep_{type} : - >)+)+$$

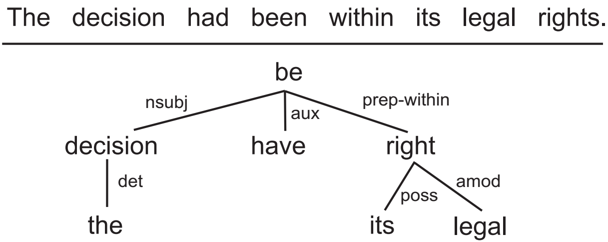
Preprocessing the Input

- Tokenize text $\rightsquigarrow$ lexical tokens (words)
- Lematize tokens $\rightsquigarrow$ lemmas
- Part-of-Speech Tagging $\rightsquigarrow$ POSs
- Extract Word Specificity from precalculated Indices (IDF, Entropy)
- Compute the Meaning of Words $\rightsquigarrow$ WordNet Sense — LSA Vector
- Syntactic Parsing $\rightsquigarrow$ dependency relations between words

A Shallow Representation of Meaning

SR: (*Word*, *Lemma*, *POS*, *Specificity*, WN-SENSE|LSA-Vector,
$(< - : dep_{type} : dep_{mod} > | < dep_{head} : dep_{type} : - >)+)+$

Extracting Dependencies



Outline

A Simple Method to Measure Similarity of Texts

Compute the degree of token overlap between the texts

Text A:	Peter went to Seattle last Thursday .
Text B:	Last Thursday, my friend Peter flew to Seattle for a business meeting .

- Number of common tokens = 6 (including punctuation)
- Average number of tokens $= \frac{7(TextA)+14(TextB)}{2} = 10.5$
- Similarity Score: $Sim = 6/10.5 = 0.57$
- Paraphrasing: Is Sim $\geq$ Threshold?
- Learn optimum threshold $\leftrightarrow$ Maximum accuracy on training

A Simple Method to Measure Similarity of Texts

Compute the degree of token overlap between the texts

Text A:	Peter went to Seattle last Thursday .
Text B:	Last Thursday, my friend Peter flew to Seattle for a business meeting .

- Number of common tokens = 6 (including punctuation)
- Average number of tokens $= \frac{7(TextA)+14(TextB)}{2} = 10.5$
- Similarity Score: $Sim = 6/10.5 = 0.57$
- Paraphrasing: Is Sim $\geq$ Threshold?
- Learn optimum threshold $\hookleftarrow$ Maximum accuracy on training

Our process:

Step 1) Find all distinct pairs

Step 2) Count the pairs (or do a weighted sum)

Step 3) Normalize (use average or maximum length)

Decisions to Consider (when counting common tokens)

- Ignore Puntuation I am not a business man. I am a business, man.

Decisions to Consider (when counting common tokens)

- Ignore Puntuation — I am not a business man. I am a business, man.
- Consider only Content Words or Ignore Stop-Words

Text A:	John is flying from Seattle.
Text B:	John is flying to Seattle.

Decisions to Consider (when counting common tokens)

- Ignore Puntuation — I am not a business man. I am a business, man.
- Consider only Content Words or Ignore Stop-Words

Text A:	John is flying from Seattle.
Text B:	John is flying to Seattle.

- Compare base form

Text A:	The children are playing in the courtyard.
Text B:	The child was playing in the courtyard.

Decisions to Consider (when counting common tokens)

- Ignore Puntuation   I am not a business man. I am a business, man.

- Consider only Content Words or Ignore Stop-Words

Text A:	John is flying from Seattle.
Text B:	John is flying to Seattle.

- Compare base form

Text A:	The children are playing in the courtyard.
Text B:	The child was playing in the courtyard.

- Ignore Case

Text A:	People were having a good time.	**Text A:**	They made US proud.
Text B:	Most people were having a good time.	**Text B:**	They made us proud.

Decisions to Consider (when counting common tokens)

- Ignore Puntuation   I am not a business man. I am a business, man.
- Consider only Content Words or Ignore Stop-Words

Text A:	John is flying from Seattle.
Text B:	John is flying to Seattle.

- Compare base form

Text A:	The children are playing in the courtyard.
Text B:	The child was playing in the courtyard.

- Ignore Case

Text A:	People were having a good time.	**Text A:**	They made US proud.
Text B:	Most people were having a good time.	**Text B:**	They made us proud.

- Compare with POS

Text A:	Trees line the riverbank.	**Text A:**	They had a pleasant walk in the park.
Text B:	The riverbank ends the line of trees.	**Text B:**	They pleasantly walked in the park.

Decisions to Consider (when counting common tokens)

- Ignore Puntuation — I am not a business man. I am a business, man.
- Consider only Content Words or Ignore Stop-Words

Text A:	John is flying from Seattle.
Text B:	John is flying to Seattle.

- Compare base form

Text A:	The children are playing in the courtyard.
Text B:	The child was playing in the courtyard.

- Ignore Case

Text A:	People were having a good time.	**Text A:**	They made US proud.
Text B:	Most people were having a good time.	**Text B:**	They made us proud.

- Compare with POS

Text A:	Trees line the riverbank.	**Text A:**	They had a pleasant walk in the park.
Text B:	The riverbank ends the line of trees.	**Text B:**	They pleasantly walked in the park.

- Unigram versus Bigram overlap - pair bigrams of tokens

Decisions to Consider (when counting common tokens)

- Ignore Puntuation — I am not a business man. I am a business, man.
- Consider only Content Words or Ignore Stop-Words

Text A:	John is flying from Seattle.
Text B:	John is flying to Seattle.

- Compare base form

Text A:	The children are playing in the courtyard.
Text B:	The child was playing in the courtyard.

- Ignore Case

Text A:	People were having a good time.	**Text A:**	They made US proud.
Text B:	Most people were having a good time.	**Text B:**	They made us proud.

- Compare with POS

Text A:	Trees line the riverbank.	**Text A:**	They had a pleasant walk in the park.
Text B:	The riverbank ends the line of trees.	**Text B:**	They pleasantly walked in the park.

- Unigram versus Bigram overlap - pair bigrams of tokens
- Weighting, Normalization

Type of a Lexical Token (Unigram versus Bigram)

Text: To be or not to be

Unigrams

Number of tokens $= 6$

Number of token types $= 4$

Frequency of token type "be" $= 2$

Bigrams

Number of bigrams $= 5$

Number of bigram types $= 4$

Frequency of bigram type "to be" $= 2$

Local and Global Weighting Schemas

$$w_{binary}(i,j) = \begin{cases} 1 & \text{if} \quad i \in j \\ 0 & \text{if} \quad i \notin j \end{cases}$$

i = type of a lexical token

j = a text instance or a document

D = a collection of documents

tf_{ij} = frequency of i in j

Local Weighting

$$lw_{frequency}(i,j) = \begin{cases} tf_{ij} & \text{if} \quad i \in j \\ 0 & \text{if} \quad i \notin j \end{cases}$$

$$lw_{logf}(i,j) = log[lw_{frequency}(i,j) + 1]$$

Global Weighting

$$gw_{entropy}(i) = 1 + \sum_j \frac{p_{ij} log_2(p_{ij})}{log_2(n)}$$

$$\text{,where } p_{ij} = \frac{tf_{ij}}{\sum_{k \in D} tf_{ik}}$$

$$gw_{idf}(i) = log \frac{|D|}{\sum_{j \in D} w_{binary}(i,j)}$$

Weighting Schemas in Semantic Similarity Assessment

$$weight(token) = weight_{local}(token) * weight_{global}(token)$$

Local Weighting

- We use only binary (no local weight) and frequency
- For binary we compare the sets of token types (or n-gram types)
- For frequency we compare the sets of tokens (or n-grams)

Global Weighting

- We use binary (no global weight), entropy and idf
- Entropy - available in the LSA space (built from TASA corpus)
- IDF - we build our own IDF index from Wikipedia

Understanding the Results

- We show accuracy and precision ...

  ... on both training and testing data

- We compare several methods in a graph (around 8 methods/graph)

- $MethodName = (ST|OP) - (P|W|C|S)(W|B|P)(C|I)(U|B)(I|E|N)(F|N)$

(ST|OP) = Stanford Processing | OpenNLP Processing

(P|W|C|S) = Punctuation | Words Only | Content Words | No Stop-Words

(W|B|P) = Compare Words | Lemmas | Lemmas with POS

(C|I) = Case Sensitive | Case Insensitive

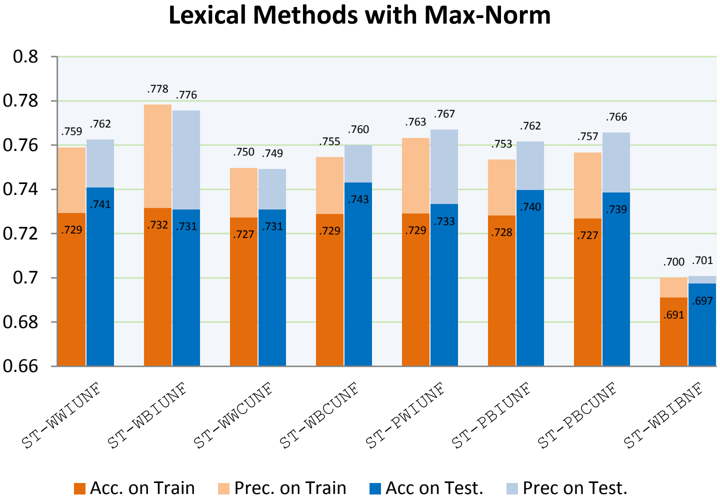
(U|B) = Unigrams | Bigrams

(I|E|N) = Global Weighting (IDF | Entropy | NoWeight)

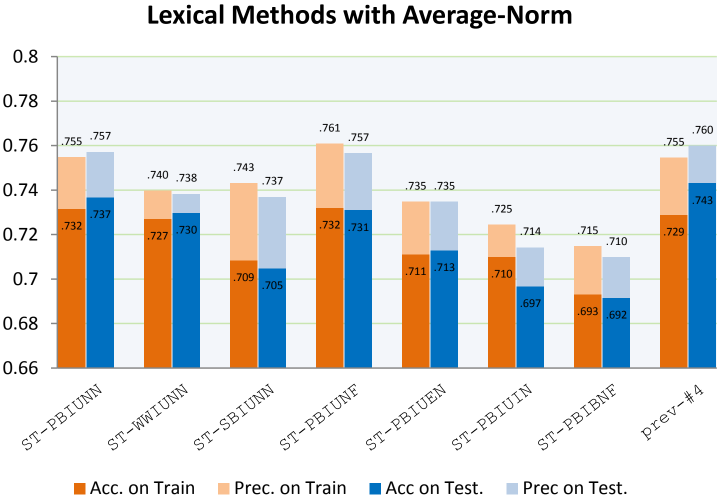
(F|N) = Local Weighting (Frequency | NoWeight)

Example: ST-W.B.I.U.N.F

Results on Lexical Methods - 1



Results on Lexical Methods - 2



Outline

Motivation for Word-to-Word Similarity Metrics

Text A: York had no problem with MTA's insisting the decision to shift funds had been within its legal rights.

Text B: York had no problem with MTA's saying the decision to shift funds was within its powers.

Motivation for Word-to-Word Similarity Metrics

Text A: York had no problem with MTA's insisting the decision to shift funds had been within its legal rights.

Text B: York had no problem with MTA's saying the decision to shift funds was within its powers.

We use **WordNet Similarity** and **LSA**-based metrics

insisting versus saying

W2W Metric	insist ⇔ say
WNS Path	0.333
WNS Lin	0.594
WNS Lch	0.670
WNS HSO	0.375
LSA	0.126

Motivation for Word-to-Word Similarity Metrics

Text A:	York had no problem with MTA's insisting the decision to shift funds had been within its legal rights.
Text B:	York had no problem with MTA's saying the decision to shift funds was within its powers.

We use **WordNet Similarity** and **LSA**-based metrics

insisting versus saying

W2W Metric	insist ⇔ say
WNS Path	0.333
WNS Lin	0.594
WNS Lch	0.670
WNS HSO	0.375
LSA	0.126

- first, pair identical tokens
- then, pair words on $W2W \geq Th_{sim}$
- greedy vs. optimal matching
- Weight-sum the pairs on their W2W metric

Greedy versus Optimal Matching

Text A:	My pet enjoys playing with your dog.
Text B:	My cat likes to play with your pet.

Greedy versus Optimal Matching

Text A:	My pet enjoys playing with your dog.
Text B:	My cat likes to play with your pet.

Greedy matching - Find the closest word

$pet_A \Leftrightarrow pet_B$ $dog_A \Leftrightarrow cat_B$

Greedy versus Optimal Matching

Text A:	My pet enjoys playing with your dog.
Text B:	My cat likes to play with your pet.

Greedy matching - Find the closest word

$pet_A \Leftrightarrow pet_B$ $\quad$ $dog_A \Leftrightarrow cat_B$

Optimal matching - Find the optimal/correct pairing

- search for the best overall matching score (i.e. sum of all matched pairs)

$pet_A \Leftrightarrow cat_B$ $\quad$ $dog_A \Leftrightarrow pet_B$

Greedy versus Optimal Matching

Text A: My pet enjoys playing with your dog.
Text B: My cat likes to play with your pet.

Greedy matching - Find the closest word

$pet_A \Leftrightarrow pet_B$ $dog_A \Leftrightarrow cat_B$

Optimal matching - Find the optimal/correct pairing

- search for the best overall matching score (i.e. sum of all matched pairs)

$pet_A \Leftrightarrow cat_B$ $dog_A \Leftrightarrow pet_B$

The Assignment Problem

- solvable in polynomial time (The Hungarian Algorithm)

Computing Similarity for W2W Methods

Symmetric Similarity ($A \Leftrightarrow B$)

$$Sim_{W2W}(A,B) = \frac{2 * \sum_{w_A \in A, w_B \in B (paired)} \frac{weight(w_A) + weight(w_B)}{2} W2W(w_A, w_B)}{\sum_{w \in A} weight(w) + \sum_{w \in B} weight(w)} \quad (1)$$

Computing Similarity for W2W Methods

Symmetric Similarity ($A \Leftrightarrow B$)

$$Sim_{W2W}(A,B) = \frac{2 * \sum_{w_A \in A, w_B \in B (paired)} \frac{weight(w_A) + weight(w_B)}{2} W2W(w_A, w_B)}{\sum_{w \in A} weight(w) + \sum_{w \in B} weight(w)} \quad (1)$$

Normalization on Maximum Length (Max-Norm) ($A \Leftrightarrow B$)

$$Sim_{W2W}(A,B) = \frac{\sum_{w_A \in A, w_B \in B (paired)} \frac{weight(w_A) + weight(w_B)}{2} W2W(w_A, w_B)}{Max(\sum_{w \in A} weight(w), \sum_{w \in B} weight(w))} \quad (2)$$

Computing Similarity for W2W Methods

Symmetric Similarity $(A \Leftrightarrow B)$

$$Sim_{W2W}(A,B) = \frac{2 * \sum_{w_A \in A, w_B \in B(paired)} \frac{weight(w_A)+weight(w_B)}{2} W2W(w_A, w_B)}{\sum_{w \in A} weight(w) + \sum_{w \in B} weight(w)} \quad (1)$$

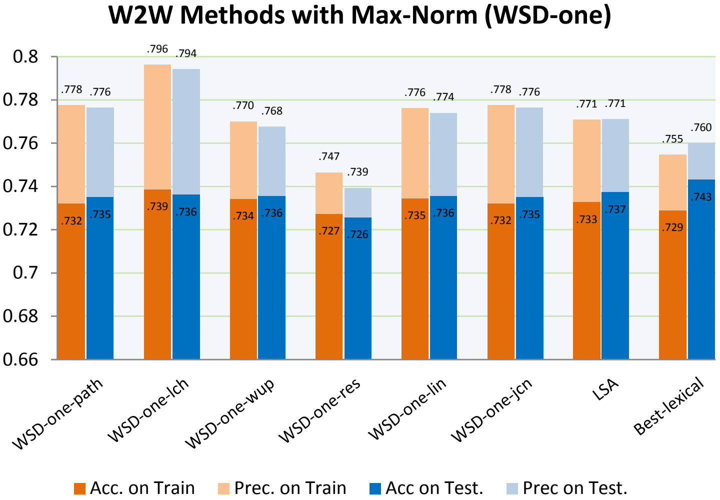
Normalization on Maximum Length (Max-Norm) $(A \Leftrightarrow B)$

$$Sim_{W2W}(A,B) = \frac{\sum_{w_A \in A, w_B \in B(paired)} \frac{weight(w_A)+weight(w_B)}{2} W2W(w_A, w_B)}{Max(\sum_{w \in A} weight(w), \sum_{w \in B} weight(w))} \quad (2)$$

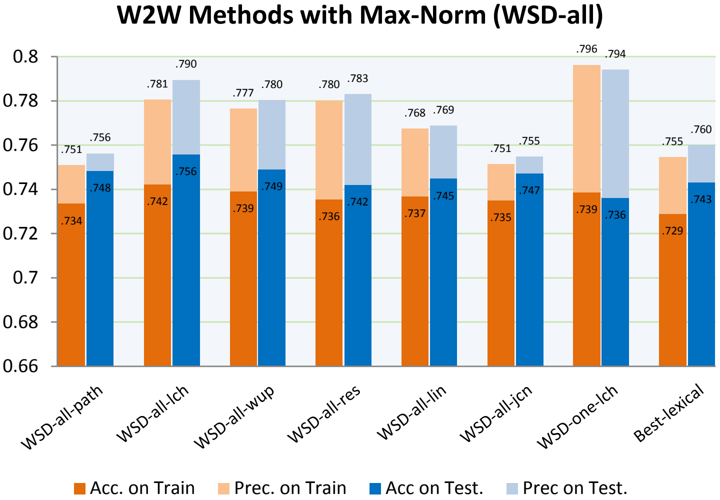
Asymmetric Similarity $(A \Rightarrow B)$

$$Sim_{W2W}(A,B) = \frac{\sum_{w_A \in A, w_B \in B(paired)} \frac{weight(w_A)+weight(w_B)}{2} W2W(w_A, w_B)}{\sum_{w \in B} weight(w)} \quad (3)$$

Results on W2W Methods - 1



Results on W2W Methods - 2



Extra Work Detailed in Chapter 4

Evaluate WordNet Relatedness Measures

- Experiment with the ULPC dataset
- Compare between using: all senses vs. first sense of words
- Evaluate the IDF weighting schema

Extra Work Detailed in Chapter 4

Evaluate WordNet Relatedness Measures

- Experiment with the ULPC dataset
- Compare between using: all senses vs. first sense of words
- Evaluate the IDF weighting schema

Evaluate LSA vectorial-based metrics

- Experiment on MSR, ULPC and PKA datasets
- In PKA we work with paragraphs instead of sentences
- Compare between using different local and global weighting

Outline

Using Dependency Relations

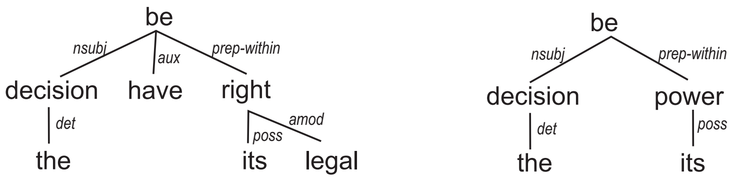
Text A:	The man chased the dog.
Text B:	The man was chased by the dog.

Text A: man is subject of chase

Text B: dog is subject of chase

Using Dependency Relations

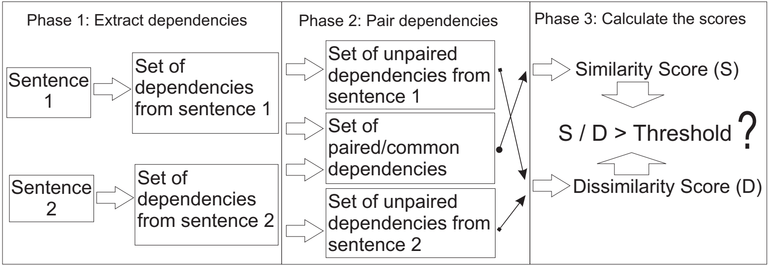
The decision had been within its legal rights. The decision was within its powers.



Paired Dependencies:
det(decision, the) = det(decision, the)
nsubj(be, decision) = nsubj(be, decision)
poss(power, its) = poss(right, its)
prep_within(be, power) = prep_within(be, right)

Unpaired Dependencies/Sentence 1:
aux(be, had)
amod(right-n, legal-a)
Unpaired Dependencies/Sentence 2:
EMPTY

Computing Dep Similarity Score



$$sim(S_1, S_2) = \sum_{d_1 \in S_1} max_{d_2 \in S_2^*} [d2dSim(d_1, d_2)]$$

$$diss(S_1, S_2) = \sum_{d_1 \in unpS_1} weight(d_1) + \sum_{d_2 \in unpS_2} weight(d_2)$$

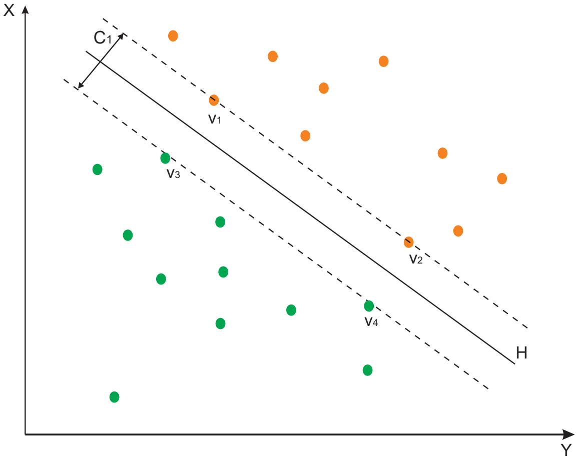
$$Sim_{dep} = sim(S_1, S_2)/diss(S_1, S_2)$$

Results with Dependency-based Methods (on MSR)

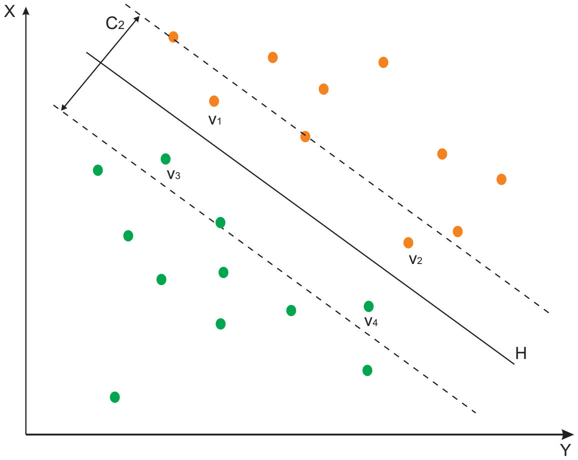
System	Acc.	Prec.	Recall	F-score
Uniform baseline	0.6649	0.6649	1.0000	0.7987
Random baseline (Corley&Mihalcea'05)	0.5130	0.6830	0.5000	0.5780
Lexical baseline (Zhang&Patrick'05)	0.7230	0.7880	0.7980	0.7930
Corley and Mihalcea (2005)	0.7150	0.7230	0.9250	0.8120
Qiu (2006)	0.7200	0.7250	0.9340	0.8160
Rus (2008) - average	0.7061	0.7207	0.9111	0.8048
Simple dep. overlap (Minipar)	0.6939	0.7109	0.9093	0.7979
Simple dep. overlap (Stanford)	0.6823	0.7064	0.8936	0.7890
Optimum results (Minipar)	0.7206	0.7404	0.8928	0.8095
Optimum results (Stanford)	0.7101	0.7270	0.9032	0.8056
No word semantics (Minipar)	0.7038	0.7184	0.9119	0.8037
No word semantics (Stanford)	0.7032	0.7237	0.8954	0.8005
No dependency weighting (Minipar)	0.7177	0.7378	0.8928	0.8079
No dependency weighting (Stanford)	0.7067	0.7265	0.8963	0.8025
No penalty for extra info (Minipar)	0.7067	0.7275	0.8936	0.8020
No penalty for extra info (Stanford)	0.7032	0.7138	0.9241	0.8055

Outline

Principle of Support Vector Machines



Principle of Support Vector Machines



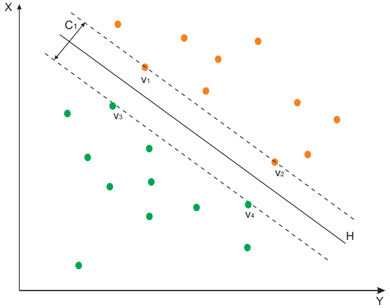
Kernels for Support Vector Machines

What if data is not linearly separable?

Kernels for Support Vector Machines

What if data is not linearly separable?

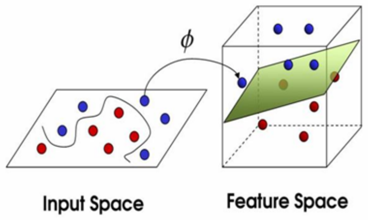
- SVM rely only on the proximity between points ⇒ Kernel functions
- The linear kernel: $K_{linear}(x, y) = (x \cdot y)$



Kernels for Support Vector Machines

What if data is not linearly separable?

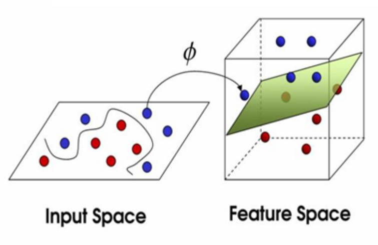
- SVM rely only on the proximity between points ⇒ Kernel functions
- The linear kernel: $K_{linear}(x, y) = (x \cdot y)$
- Valid kernels map data in new spaces, of any number of desired dimensions



Kernels for Support Vector Machines

What if data is not linearly separable?

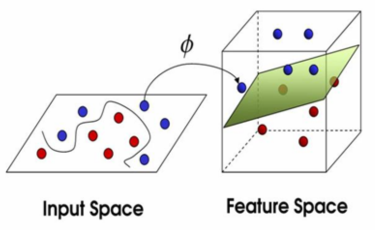
- SVM rely only on the proximity between points ⇒ Kernel functions
- The linear kernel: $K_{linear}(x, y) = (x \cdot y)$
- Valid kernels map data in new spaces, of any number of desired dimensions
- No need to represent data in the new space. Use only the kernel function



Kernels for Support Vector Machines

What if data is not linearly separable?

- SVM rely only on the proximity between points ⇒ Kernel functions
- The linear kernel: $K_{linear}(x, y) = (x \cdot y)$
- Valid kernels map data in new spaces, of any number of desired dimensions
- No need to represent data in the new space. Use only the kernel function



Classic Kernel functions

polynomial	$K_{poly}(x, y) = (x \cdot y + coef)^d$
radial basis	$K_{rad}(x, y) = exp(-\gamma \|\|x - y\|\|)^2$
two layer sigmoid	$K_{sig}(x, y) = tanh(\gamma xy + coef)$

String Kernels

Why kernels in NLP

- Language processing tasks are highly dimensional

  $\Rightarrow$ Every word counts (and is often used as a dimension)

- Kernel functions are very helpful in dealing with highly dimensional problems

  $\Rightarrow$ String kernels define a dimensions for each word in the vocabulary

- A classic string kernel

  $K_{string}(A, B) =$number of common words between A and B

String Kernels in Semantic Similarity Assessment

- Data points are instances of pairs (there are two sentences per pair)
- How to measure the proximity of two instances, A and B

C_A / D_A = set of common /different words in instance A
C_B / D_B = set of common/different words in instance B

$$K_{simm}(A, B) = \text{number of common words between } C_A \text{ and } C_B$$

$$K_{diss}(A, B) = \text{number of common words between } D_A \text{ and } D_B$$

$$K_{DisSim}(A, B) = K_{diss}(A, B) + K_{simm}(A, B)$$

- What do we compare

we use words, lemmas, parts-or-speech or dependency paths

How does a dissimilarity kernel work?

Text A_1: Mary went to the doctor yesterday.
Text A_2: I saw Mary going to the doctor the other day.

Text B_1: Josh bought some shoes from the mall.
Text B_2: I saw Josh buying some shoes at the mall the other day.

How does a dissimilarity kernel work?

Text A_1: Mary went to the doctor yesterday.
Text A_2: I saw Mary going to the doctor the other day.

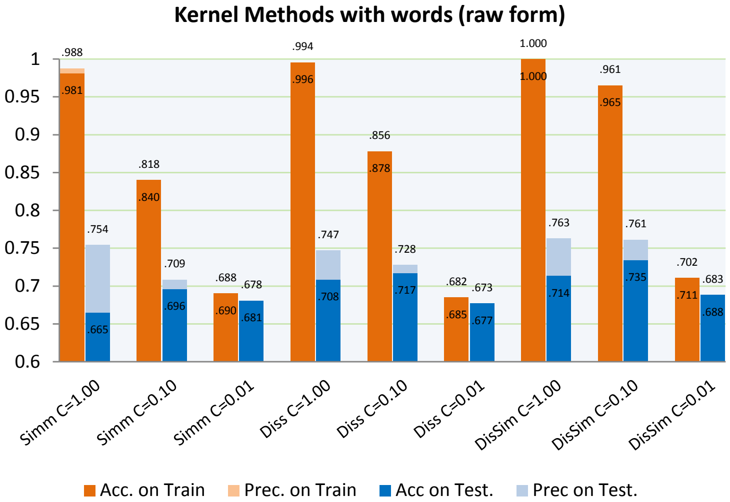
Text B_1: Josh bought some shoes from the mall.
Text B_2: I saw Josh buying some shoes at the mall the other day.

D_A = (yesterday, I, saw, the, other, day)

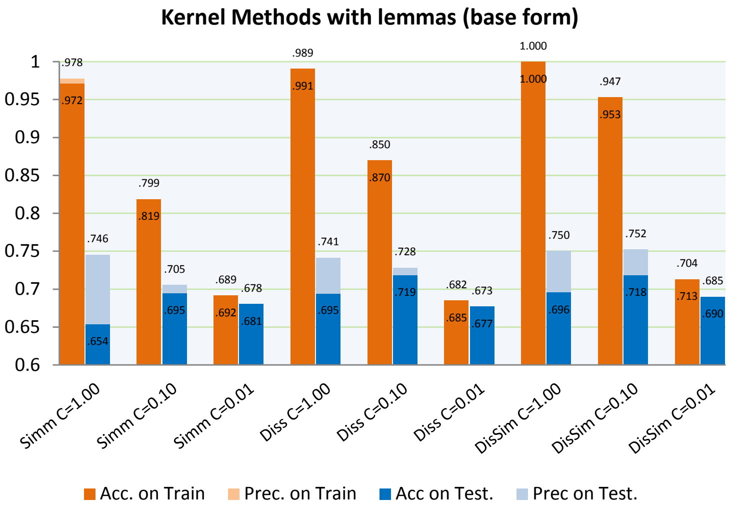
D_B = (I, saw, the, other, day)

$$\implies K_{diss}(A, B) = 5$$

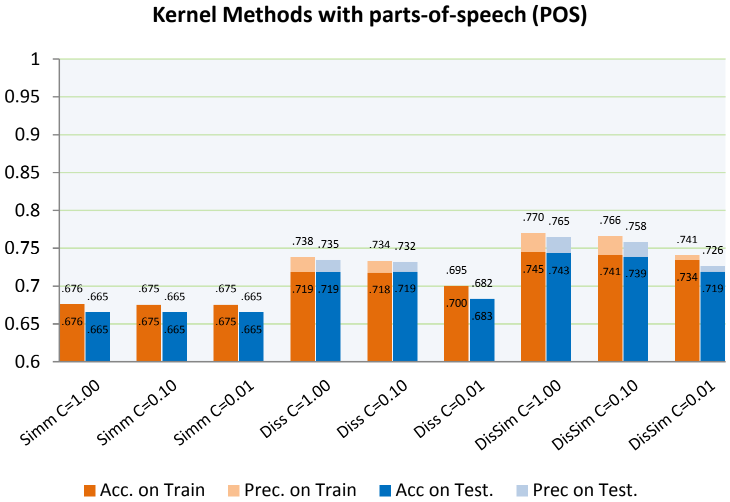
Results on Kernel Methods - 1



Results on Kernel Methods - 2



Results on Kernel Methods - 3



Outline

Conclusions

- We explored the role of various levels of linguistic information (lexical, syntactic and semantic) on the task of semantic similarity assessment.
- We showed that simple methods (i.e. token overlap), are much more complex than they are usually addressed in the literature, and we addressed this problem by proposing a framework that allows the exploration of a large parameter space for simple overlap methods.
- We explored a range of methods from simple token overlap to optimum word-based similarity methods to kernel-based methods.
- There is need for better corpora to study the task of semantic similarity assessment, as existing corpora have significant limitations and can be misleading with respect to the potential of various methods addressing the task of semantic similarity.

Future Work

- improve the kernel methods $\rightarrow$ asymmetric similarity
- explore more ways to use the framework
- do qualitative analysis on the output of methods

Publications

Journal Publications

Lintean, M., & Rus, V. (2010). Paraphrase Identification Using Weighted Dependencies and Word Semantics. *Informatica, An International Journal of Computing and Informatics*.

Conference Proceedings

- Lintean, M, & Rus, V. (2011). Dissimilarity Kernels for Paraphrase Identification. *Proceedings of the 24th International Florida Artificial Intelligence Research Society Conference*. Palm Beach, FL.
- Lintean, M, & Rus, V. (2010). The Role of Local and Global Weighting in Assessing The Semantic Similarity of Texts using Latent Semantic Analysis. *Proceedings of the 23rd International Florida Artificial Intelligence Research Society Conference*. Daytona Beach, FL.
- Lintean, M, Rus, V., Graesser, A., & McNamara, D. (2009). Assessing Student Paraphrases Using Lexical Semantics and Word Weighting. *Proceedings of the 14th International Conference on Artificial Intelligence in Education*. Brighton, UK.
- Lintean, M, & Rus, V. (2010). Paraphrase Identification Using Weighted Dependencies and Word Semantics. *Proceedings of the 22nd International Florida Artificial Intelligence Research Society Conference*. Sanibel Island, FL.
- Lintean, M, & Rus, V. (2010). Using Dependency Relations to Decide Paraphrasing. *Proceedings of the Society for Text and Discourse Conference*. Memphis, TN.

Outline

Accuracy, Precision, Recall

Confusion Matrix

Actual \ Predicted	False	True
False	A	B
True	C	D

$Accuracy = \frac{A+D}{A+B+C+D}$

$Precision = \frac{D}{B+D}$

$Recall = \frac{D}{C+D}$

$Actual_{true} > Actual_{false} \quad \Rightarrow \quad Accuracy \leq Precision$

$Predicted_{true} < Actual_{true}$

Formulas of Lexical Overlap

Weighted Similarity

$$WSim(A,B) = \frac{2 * \sum_{w \in C} [weight_{global}(w) * weight_{local}(w)]}{\sum_{w \in A \uplus B} [weight_{global}(w) * weight_{local}(w)]} \tag{4}$$

Asymmetric Similarity

$$WSim(A,B) = \frac{\sum_{w \in C} [weight_{global}(w) * weight_{local}(w)]}{\sum_{w \in B} [weight_{global}(w) * weight_{local}(w)]} \tag{5}$$

Normalization on Maximum Length

$$WSim(T_1,T_2) = \frac{\sum_{w \in C} [weight_{global}(w) * weight_{local}(w)]}{Max(\sum_{w \in A | w \in B} [weight_{global}(w) * weight_{local}(w)])} \tag{6}$$

Results on Lexical Methods (in tabular view)

Lexical methods on MSR, with Stanford parsing and Max-Norm

		Performance on Train			Performance on Test		
Method	Threshold	Acc.	Prec.	Recall	Acc.	Prec.	Recall
W.W.I.U.N.F.	.4828	.7294	.7589	.8783	.7409	.7624	.8867
W.B.I.U.N.F.	.5263	.7316	.7783	.8427	.7310	**.7756**	.8378
W.W.C.U.N.F.	.4615	.7274	.7497	.8954	.7310	.7491	.8954
W.B.C.U.N.F.	.5000	.7289	.7546	.8870	**.7432**	.7600	.8971
P.W.I.U.N.F.	.5238	.7291	.7632	.8685	.7333	.7669	.8605
P.B.I.U.N.F.	.5238	.7282	.7533	.8885	.7397	.7616	.8858
P.B.C.U.N.F.	.5238	.7269	.7567	.8780	.7386	.7656	.8745
W.B.I.B.N.F.	.1818	.6911	.7001	.9495	.6974	.7007	.9512

Mercer's Theorem

A valid kernel must respect Mercer's condition

$$\int \int K(x, y)g(x)g(y)dxdy \geq 0 \tag{7}$$

A symmetric continuos, non-negative definite function

$$\sum_{i=1}^{n} sum_{j=1}^{n} K(x_i, x_j)c_i c_j \geq 0 \tag{8}$$

The Vapnik Chervonenkis (VC) dimension

The Empirical Risk

$$R_{emp}(\alpha) = \frac{1}{2l} \sum_{i=1}^{l} |y_i - f(x_i, \alpha)| \tag{9}$$

The Calculated Risk Bound

$$R(\alpha) = \int \frac{1}{2} |y - f(x, \alpha)| dP(x, y) \tag{10}$$

The VC dimension (h > 0)

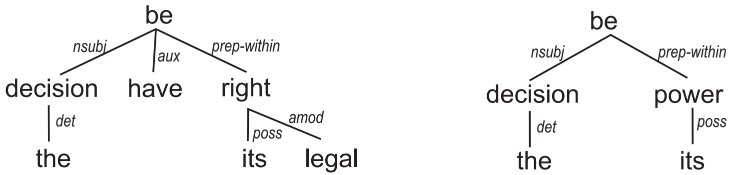
$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\frac{h(log(2l/h) + 1) - log(\mu/4)}{l}} \tag{11}$$

Dependency Based Kernels

- We also experimented with Dependecy-based kernels

The decision had been within its legal rights.

The decision was within its powers.



Paired Dependencies:
det(decision, the) = det(decision, the)
nsubj(be, decision) = nsubj(be, decision)
poss(power, its) = poss(right, its)
prep_within(be, power) = prep_within(be, right)

Unpaired Dependencies/Sentence 1:
aux(be, had)
amod(right-n, legal-a)
Unpaired Dependencies/Sentence 2:
EMPTY

Common subpaths:

$be \rightarrow decision \rightarrow the$
$be \rightarrow decision; decision \rightarrow the$
$its; be; decision; the$

$(be \rightarrow power) \iff (be \rightarrow right)$
$(power \rightarrow its) \iff (right \rightarrow its)$
$(power) \iff (right)$