Recognition of Literal and Non-Literal Use of Idiomatic Expressions

Linlin Li and Caroline Sporleder

MMCI / Computational Linguistics, Saarland University

{linlin,csporled}@coli.uni-sb.de

Computational Approaches to Creative Language
June 15, 2010

Outline

- Introduction
- Related Work
- Linguistic Observation
- Models
  - Lexical Chain Model
  - Cohesion Graph Model
  - Supervised Model
    - Formal Definition of Different Cohesion Features
  - Bootstrapping Model
  - Gausian Mixture Model *
- Experiments
- Conclusion

Why is Non-Literal Language a Problem?

Examples of Non-Literal Language

Dissanayake said that Kumaratunga was "playing with fire" after she accused military's top brass of interfering in the peace process. Kumaratunga has said in an interview she would not tolerate attempts by the army high command to sabotage her peace moves. A defence analyst close to the government said Kumaratunga had spoken a "load of rubbish" and the security forces would not take kindly to her disparaging comments about them.

Why is Non-Literal Language a Problem?

Examples of Non-Literal Language

Dissanayake said that Kumaratunga was "playing with fire" after she accused military's top brass of interfering in the peace process. Kumaratunga has said in an interview she would not tolerate attempts by the army high command to sabotage her peace moves. A defence analyst close to the government said Kumaratunga had spoken a "load of rubbish" and the security forces would not take kindly to her disparaging comments about them.

Non-Literal Expressions (idioms, metaphors etc.) ...

- occur frequently in language
- often behave idiosyncratically
- have to be recognised automatically to be analysed and interpreted in an appropriate way

Dealing with Idioms

Most previous research:

- automatic idiom extraction methods (type-based classification)

But:

- doesn't work for creative language use
- potentially idiomatic expressions can be used in literal sense

Literal Usage

(1) *Somehow I always end up spilling the beans all over the floor and looking foolish when the clerk comes to sweep them up.*

(2) *Grilling outdoors is much more than just another dry-heat cooking method. It's the chance to play with fire, satisfying a primal urge to stir around in coals.*

Dealing with Idioms

Most previous research:

- automatic idiom extraction methods (type-based classification)

But:

- doesn't work for creative language use
- potentially idiomatic expressions can be used in literal sense

Literal Usage

(1) *Somehow I always end up spilling the beans all over the floor and looking foolish when the clerk comes to sweep them up.*

(2) *Grilling outdoors is much more than just another dry-heat cooking method. It's the chance to play with fire, satisfying a primal urge to stir around in coals.*

⇒ **Idioms have to be recognised in discourse context!** (token-based classification)

Token-based Idiom Classification

Previous Approaches:

- Katz and Giesbrecht (2006): supervised machine learning (k-nn), vector space model
- Birke and Sarkar (2006): bootstrapping from seed lists
- Cook et al. (2007), Fazly et al. (to appear): unsupervised, predict non-literal if idiom is in canonical form ($\approx$ dictionary form)

$\Rightarrow$ limited contribution of discourse context

How do you know whether an expression is used idiomatically?

Literal Usage

Grilling outdoors is much more than just another dry-heat cooking method. It's the chance to play with fire, satisfying a primal urge to stir around in coals.

How do you know whether an expression is used idiomatically?

Literal Usage

Grilling outdoors is much more than just another dry-heat cooking method. It's the chance to play with fire, satisfying a primal urge to stir around in coals.

Literally used expressions typically exhibit lexical cohesion with the surrounding discourse (e.g. participate in lexical chains of semantically related words).

How do you know whether an expression is used idiomatically?

Non-Literal Usage

Dissanayake said that Kumaratunga was "playing with fire" after she accused military's top brass of interfering in the peace process. Kumaratunga has said in an interview she would not tolerate attempts by the army high command to sabotage her peace moves. A defence analyst close to the government said Kumaratunga had spoken a "load of rubbish" and the security forces would not take kindly to her disparaging comments about them.

Non-Literally used expressions typically do **not** participate in cohesive chains.

Limitations of the Cohesion-Based Approach

Literal Use without Lexical Chain

Chinamasa compared McGown's attitude to morphine to a child's attitude to playing with fire – a lack of concern over the risks involved.

Non-Literal Use with Lexical Chain

Saying that the Americans were "playing with fire" the official press speculated that the "gunpowder barrel" which is Taiwan might well "explode" if Washington and Taipei do not put a stop to their "incendiary gesticulations."

⇒ Both cases are relatively rare

A Cohesion-based Approach to Idiom Detection

Identifying Idiomatic Usage

Are there (strong) cohesive ties between the component words of the idiom and the context?

- Yes $\Rightarrow$ literal usage
- No $\Rightarrow$ non-literal usage

(cf. Hirst and St-Onge's (1998) work on detecting malapropisms)

We need:

- a measure of semantic relatedness
- a method for modelling lexical cohesion:
  - lexical chains
  - cohesion graphs

We have to model non-classical relations (e.g. *fire - coals*, *sweep up - spill*, *ice - freeze*) and world knowledge (*Wayne Rooney - ball*).

⇒ distributional approaches better suited than WordNet-based ones
⇒ ideally, we need loads of up-to-date data

Normalised Google Distance (NGD) (Cilibrasi and Vitanyi, 2007)

- use search engine page counts (here: Yahoo) as proxies for word co-occurrence

$$NGD(x,y) = \frac{max\{log\ f(x), log\ f(y)\} - log\ f(x,y)}{log\ M - min\{log\ f(x), log\ f(y)\}}$$

(x, y: target words, M: total number of pages indexed)

Modelling Cohesion: Lexical Chains

Literal Use

Dad had to <u>break the ice</u> on the chicken troughs so that they could get water.

Modelling Cohesion: Lexical Chains

Literal Use

Dad had to break the ice on the chicken troughs so that they could get water.

Four Lexical Chains:

- Chain 1: *Dad*
- Chain 2: *break*
- Chain 3: *ice – water*
- Chain 4: *chicken – troughs*

Modelling Cohesion: Lexical Chains

Literal Use

Dad had to break the ice on the chicken troughs so that they could get water.

Four Lexical Chains:

- Chain 1: *Dad*
- Chain 2: *break*
- Chain 3: *ice – water*
- Chain 4: *chicken – troughs*

⇒ **Literal!**

Modelling: Lexical Chains

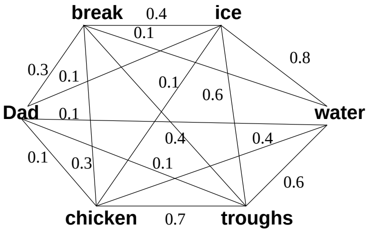
Drawbacks:

- one free parameter (similarity threshold t) for deciding when to put two words in the same chain
  $\Rightarrow$ needs to be optimised on an annotated data set (weakly supervised)
- approach is sensitive to chaining algorithm and parameter settings

Modelling Cohesion: Cohesion Graphs

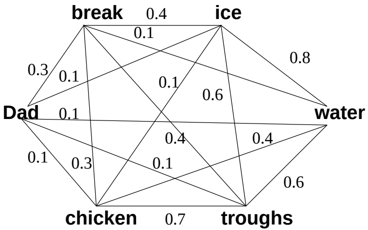
Literal Use

Dad had to break the ice on the chicken troughs so that they could get water.



Literal Use

Dad had to <u>break the ice</u> on the chicken troughs so that they could get water.



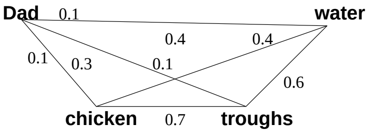
with idiom:

avg. connectivity=0.34

Modelling Cohesion: Cohesion Graphs

Literal Use

Dad had to break the ice on the chicken troughs so that they could get water.



with idiom:

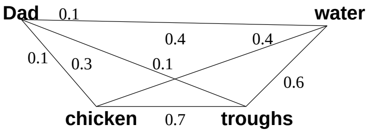
avg. connectivity=0.34

without idiom:
avg. connectivity=0.33

Modelling Cohesion: Cohesion Graphs

Literal Use

Dad had to break the ice on the chicken troughs so that they could get water.



with idiom:

avg. connectivity=0.34

without idiom:
avg. connectivity=0.33

⇒ **Literal!**

Weighting the Graph: edges

The further two tokens occur from each other, the more likely it is that their relatedness is accidental

Low Weight Edge

Next week the two diplomats will meet in an attempt to <u>break the ice</u> between the two nations. A crucial issue in the talks will be the long-running <u>water</u> dispute.

Weighting the Graph: edges

The further two tokens occur from each other, the more likely it is that their relatedness is accidental

Low Weight Edge

Next week the two diplomats will meet in an attempt to break the ice between the two nations. A crucial issue in the talks will be the long-running water dispute.

Weighting the Graph: edges

The further two tokens occur from each other, the more likely it is that their relatedness is accidental

Low Weight Edge

Next week the two diplomats will meet in an attempt to break the ice between the two nations. A crucial issue in the talks will be the long-running water dispute.

- defined in terms of the inverse of the distance δ between the two token positions id_i and id_j:

$$\lambda_{ij} = \frac{\delta(id_i, id_j)}{\sum_j \delta(id_i, id_j)}$$

Weighting the Graph: nodes

Less important tokens should be assigned less weight when modelling discourse connectivity

Low Weight Node

"Gujral will meet Sharif on Monday and discuss bilateral relations," the Press Trust of India added. The minister said Sharif and Gujral would be able to "<u>break the ice</u>" over Kashmir.

Weighting the Graph: nodes

Less important tokens should be assigned less weight when modelling discourse connectivity

Low Weight Node

"Gujral will meet Sharif on Monday and discuss bilateral relations," the Press Trust of India added. The minister said Sharif and Gujral would be able to "break the ice" over Kashmir.

- the saliency of a token for the semantic context of the text is defined on a *tf.idf*-based weighting scheme:

$$saliency(t_i) = log \frac{|D|}{|\{d : t_i \in d\}|}$$

Weighting the Graph: nodes

Less important tokens should be assigned less weight when modelling discourse connectivity

Low Weight Node

"Gujral will meet Sharif on Monday and discuss bilateral relations," the Press Trust of India added. The minister said Sharif and Gujral would be able to "break the ice" over Kashmir.

- the saliency of a token for the semantic context of the text is defined on a *tf.idf*-based weighting scheme:

$$saliency(t_i) = log \frac{|D|}{|\{d : t_i \in d\}|}$$

- weights of the nodes:

$$\beta_i = \frac{saliency(t_i)}{\sum_j saliency(t_j)}$$

More on Cohesion Graph

There is more you can do...

- Building graph on different context sizes
- Pruning the graph, delete bad connected nodes (stopwords)

Problems with the Unsupervised Classifier

Many other linguistic clues are missing

- indicative prepositions (e.g.: between, over following break the ice)

Idiomatic Usages

(1) "Gujral will meet Sharif on Monday and discuss bilateral relations," the Press Trust of India added. The minister said Sharif and Gujral would be able to break the ice over Kashmir.

Problems with the Unsupervised Classifier

Many other linguistic clues are missing

- indicative prepositions (e.g.: between, over following break the ice)

Idiomatic Usages

(1) "Gujral will meet Sharif on Monday and discuss bilateral relations," the Press Trust of India added. The minister said Sharif and Gujral would be able to break the ice over Kashmir.

(2) Next week the two diplomats will meet in an attempt to break the ice between the two nations. A crucial issue in the talks will be the long-running water dispute.

Problems with the Unsupervised Classifier

Many other linguistic clues are missing

- indicative prepositions (e.g.: between, over following break the ice)
- idiomatic usages also exhibit cohesion with their context (e.g.: break the ice co-occurs with discuss, relations, talks, diplomacy)

Idiomatic Usages

(1) "Gujral will meet Sharif on Monday and discuss bilateral relations," the Press Trust of India added. The minister said Sharif and Gujral would be able to break the ice over Kashmir.

(2) Next week the two diplomats will meet in an attempt to break the ice between the two nations. A crucial issue in the talks will be the long-running water dispute.

What do we do to overcome these problems?

More sophisticated models...

Supervised Classifier

Aims to capture more statistical information from the training data to guide your classification

$\Rightarrow$

- How to represent statistical information from the training data? $\longrightarrow$ feature selection
- How to build a statistical model to capture this information? $\longrightarrow$ model selection

Feature Design

Aim

- Phrase independent features
- Generalize across different idiomatic phrases

Features

- Semantic cohesion features
- Use normalized Google distance (Cilibrasi and Vitanyi, 2007), to model semantic cohesion

Features

- Frequency-based Features
- Semantic Cohesion Features

Frequency-based Features

- **Salient words (salW)**: identify words which are particularly *salient* for literal usage, encode the frequencies of those words in the feature vectors

$$sal_{lit}(w) = \frac{\log f_{lit}(w) \times i_{lit}(w)}{\log f_{nonlit}(w) \times i_{nonlit}(w)}$$

  ($sal_{lit}(w)$: saliency score for the class *lit*; $f_{lit}(w)$: word frequency; $i_{lit}(w)$: instance frequency)

- **Related words (relS)**: encode the frequency of the top ranked words whose semantic relatedness with the noun in the idiomatic expression are highest

Semantic Cohesion Features

Connectivity

- $x1$: the average relatedness between the target expression and context words

$$x1 = \frac{2}{|T| \times |C|} \sum_{(w_i, c_j) \in T \times C} relatedness(w_i, c_j)$$

- $x2$: the average semantic relatedness of the context words

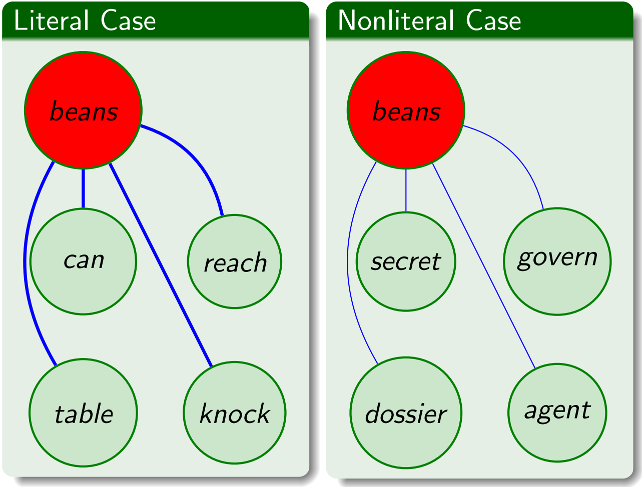
$$x2 = \frac{1}{\binom{|C|}{2}} \sum_{(c_i, c_j) \in C \times C, i \neq j} relatedness(c_i, c_j)$$

- $x3$: $x1 - x2$
- $x4$: prediction of the co-graph (Sporleder and Li, 2009)

Related Score (relS)

- $x5$: the top n relatedness scores ($n = 100$)

$$x5(k) = \max_{(w_i, c_j) \in T \times C} (k, \{relatedness(w_i, c_j)\})$$

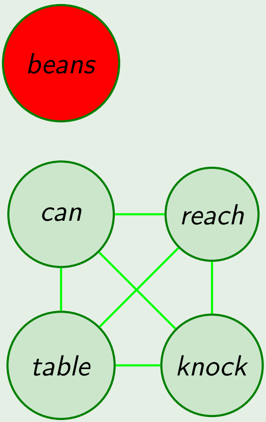


Features:

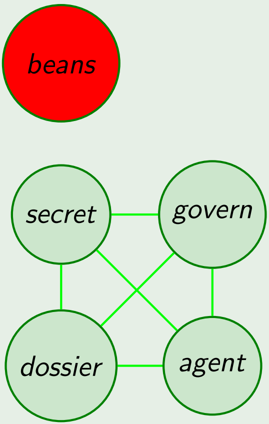
- target word connectivity (x_1)

Cohesion Features
An Example

Literal Case



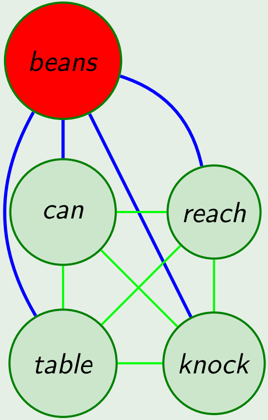
Nonliteral Case



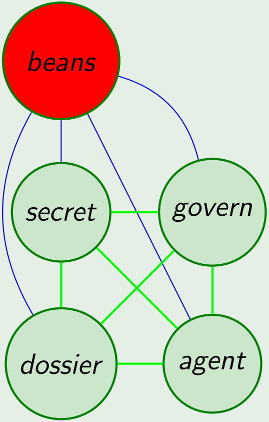
Features:

- average discourse connectivity (x_2)

Literal Case



Nonliteral Case

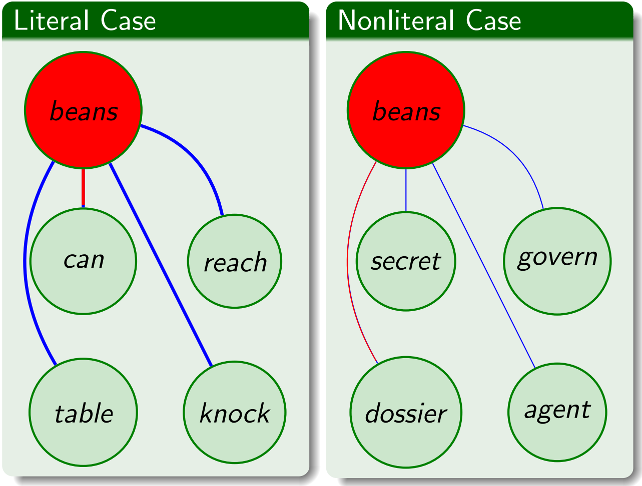


Features:

- cohesion graph ($x_1 - x_2$)

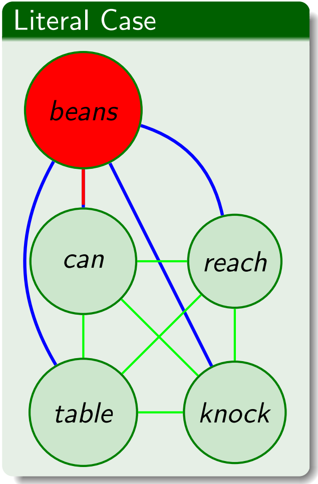
Cohesion Features

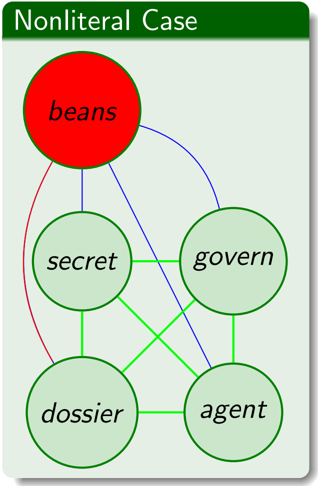
An Example



Features:

- top connected words (x_5)





Features:

- target word connectivity (x_1)
- average discourse connectivity (x_2)
- cohesion graph ($x_1 - x_2$)
- top connected words (x_5)

Model Selection

There is a wide range of different types of machine learning classifiers that you can chose from

- Depending on your problem, choose the classifier that fits the best

We choose Support Vector Machine

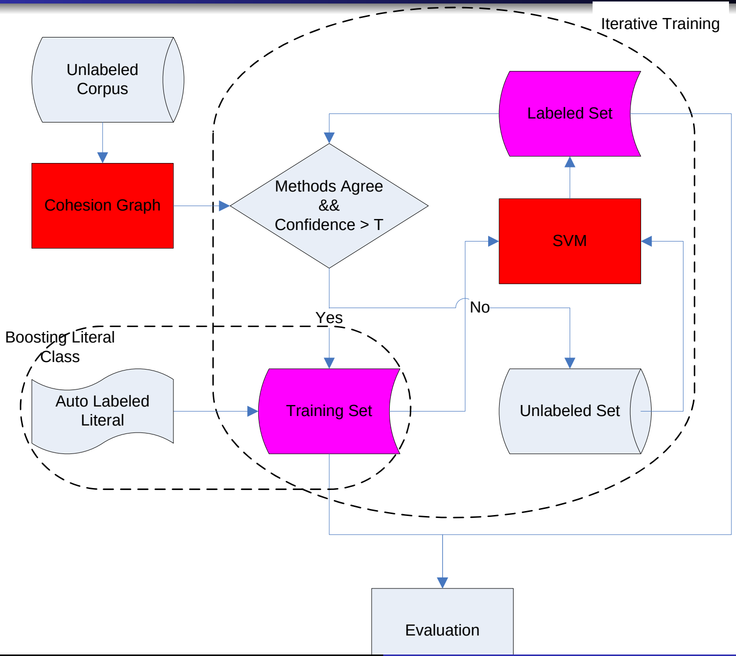
- A high dimensional numeric feature space
- SVM is fast and robust to this type of problems

Disadvantage of the Supervised Classifier

It needs a large amount of training data, which can be a huge cost!
$\Rightarrow$ Bootstrapping

- Start the classification from an unsupervised model (e.g., co-graph)
- Complement with a supervised classifier, which uses the most confident examples from the former step as training data.
- The output of this classifier goes into the first step. This process goes until convergence.

Combining the Classifiers (1)



Combining the Classifiers (2)

- Use the unsupervised classifier to label an initial training set for the supervised one
- Iteratively enlarging the training set
  - Only consider instances on whose labels both classifiers agree (reduce the noise)
  - Connectivity change of the unsupervised classifier is used as the confidence function
  - Re-training process involves re-computing the ranked lists of *salient and related words* and encode them in the feature vector

Problem: the iterative process introduce more and more imbalance in the training set

Combing the Classifiers (3)

Boosting the Literal Class

- Extract non-canonical form variants and label them as *literal* automatically
  - Change the number of the noun (*rock the boat* ⇒ *rock the boats*)
  - Change the determiner (*rock a boat*)
  - Replace the verb or noun by one of its synonyms, hypernyms or siblings from WordNet (*rock the ship*)
- Add additional literal examples during each iteration

Experiments

Data

- 17 idioms (mainly V+NP and V+PP) with literal and non-literal sense
- all (canonical form) occurrences extracted from a Gigaword corpus (3964 instances)
- five paragraphs context
- manually labelled as "literal" (862 instances) or "non-literal" (3102 instances)

Data (* = literal use is more common)

expression	literal	non-literal	all
back the wrong horse	0	25	25
bite off more than one can chew	2	142	144
bite one's tongue	16	150	166
blow one's own trumpet	0	9	9
bounce off the wall*	39	7	46
break the ice	20	521	541
drop the ball*	688	215	903
get one's feet wet	17	140	157
pass the buck	7	255	262
play with fire	34	532	566
pull the trigger*	11	4	15
rock the boat	8	470	478
set in stone	9	272	281
spill the beans	3	172	175
sweep under the carpet	0	9	9
swim against the tide	1	125	126
tear one's hair out	7	54	61
all	862	3102	3964

Results (Lexical Chain & Cohesion Graph)

	B_{Maj}	B_{Rep}	*Graph*	LC_d	LC_o
Acc	78.25	79.06	79.61	80.50	80.42
lit. Prec	-	70.00	52.21	62.26	53.89
lit. Rec	-	5.96	67.87	26.21	69.03
lit. $\mathbf{F}_{\beta=1}$	-	**10.98**	**59.02**	**36.90**	**60.53**

- $\mathbf{B_{Maj}}$: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- $\mathbf{B_{Rep}}$: predict "literal" if an idiom component word is repeated in the context
- **Graph**: cohesion graph
- $\mathbf{LC_d}$: lexical chains optimised on development set
- $\mathbf{LC_o}$: lexical chains optimised globally by oracle (upper bound for lexical chains)

	B_{Maj}	B_{Rep}	*Graph*	LC_d	LC_o
Acc	78.25	79.06	79.61	80.50	80.42
lit. Prec	-	70.00	52.21	62.26	53.89
lit. Rec	-	5.96	67.87	26.21	69.03
lit. $F_{\beta=1}$	-	**10.98**	**59.02**	**36.90**	**60.53**

- **B_{Maj}**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **B_{Rep}**: predict "literal" if an idiom component word is repeated in the context
- **Graph**: cohesion graph
- **LC_d**: lexical chains optimised on development set
- **LC_o**: lexical chains optimised globally by oracle (upper bound for lexical chains)

Results (Lexical Chain & Cohesion Graph)

	B_{Maj}	B_{Rep}	*Graph*	LC_d	LC_o
Acc	78.25	79.06	79.61	80.50	80.42
lit. Prec	-	70.00	52.21	62.26	53.89
lit. Rec	-	5.96	67.87	26.21	69.03
lit. $\mathbf{F}_{\beta=1}$	-	**10.98**	**59.02**	**36.90**	**60.53**

- **$\mathbf{B_{Maj}}$**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **$\mathbf{B_{Rep}}$**: predict "literal" if an idiom component word is repeated in the context
- **Graph**: cohesion graph
- **$\mathbf{LC_d}$**: lexical chains optimised on development set
- **$\mathbf{LC_o}$**: lexical chains optimised globally by oracle (upper bound for lexical chains)

Results (Lexical Chain & Cohesion Graph)

	B_{Maj}	B_{Rep}	*Graph*	LC_d	LC_o
Acc	78.25	79.06	79.61	80.50	80.42
lit. Prec	-	70.00	52.21	62.26	53.89
lit. Rec	-	5.96	67.87	26.21	69.03
lit. $\mathbf{F}_{\beta=1}$	-	**10.98**	**59.02**	**36.90**	**60.53**

- $\mathbf{B_{Maj}}$: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- $\mathbf{B_{Rep}}$: predict "literal" if an idiom component word is repeated in the context
- **Graph**: cohesion graph
- $\mathbf{LC_d}$: lexical chains optimised on development set
- $\mathbf{LC_o}$: lexical chains optimised globally by oracle (upper bound for lexical chains)

Results (Lexical Chain & Cohesion Graph)

	B_{Maj}	B_{Rep}	*Graph*	LC_d	LC_o
Acc	78.25	79.06	79.61	80.50	80.42
lit. Prec	-	70.00	52.21	62.26	53.89
lit. Rec	-	5.96	67.87	26.21	69.03
lit. $\mathbf{F}_{\beta=1}$	-	**10.98**	**59.02**	**36.90**	**60.53**

- $\mathbf{B_{Maj}}$: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- $\mathbf{B_{Rep}}$: predict "literal" if an idiom component word is repeated in the context
- **Graph**: cohesion graph
- $\mathbf{LC_d}$: lexical chains optimised on development set
- $\mathbf{LC_o}$: lexical chains optimised globally by oracle (upper bound for lexical chains)

Results (Lexical Chain & Cohesion Graph)

	B_{Maj}	B_{Rep}	*Graph*	LC_d	LC_o
Acc	78.25	79.06	79.61	80.50	80.42
lit. Prec	-	70.00	52.21	62.26	53.89
lit. Rec	-	5.96	67.87	26.21	69.03
lit. $F_{\beta=1}$	-	**10.98**	**59.02**	**36.90**	**60.53**

- **B_{Maj}**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **B_{Rep}**: predict "literal" if an idiom component word is repeated in the context
- **Graph**: cohesion graph
- **LC_d**: lexical chains optimised on development set
- **LC_o**: lexical chains optimised globally by oracle (upper bound for lexical chains)

Results (Lexical Chain & Cohesion Graph)

	B_{Maj}	B_{Rep}	*Graph*	LC_d	LC_o
Acc	78.25	79.06	79.61	80.50	80.42
lit. Prec	-	70.00	52.21	62.26	53.89
lit. Rec	-	5.96	67.87	26.21	69.03
lit. $\mathbf{F}_{\beta=1}$	-	**10.98**	**59.02**	**36.90**	**60.53**

- $\mathbf{B_{Maj}}$: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- $\mathbf{B_{Rep}}$: predict "literal" if an idiom component word is repeated in the context
- **Graph**: cohesion graph
- $\mathbf{LC_d}$: lexical chains optimised on development set
- $\mathbf{LC_o}$: lexical chains optimised globally by oracle (upper bound for lexical chains)

Results (Lexical Chain & Cohesion Graph)

	B_{Maj}	B_{Rep}	*Graph*	LC_d	LC_o
Acc	78.25	79.06	79.61	80.50	80.42
lit. Prec	-	70.00	52.21	62.26	53.89
lit. Rec	-	5.96	67.87	26.21	69.03
lit. $\mathbf{F}_{\beta=1}$	-	**10.98**	**59.02**	**36.90**	**60.53**

- **$\mathbf{B_{Maj}}$**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **$\mathbf{B_{Rep}}$**: predict "literal" if an idiom component word is repeated in the context
- **Graph**: cohesion graph
- **$\mathbf{LC_d}$**: lexical chains optimised on development set
- **$\mathbf{LC_o}$**: lexical chains optimised globally by oracle (upper bound for lexical chains)

Results (Lexical Chain & Cohesion Graph)

	B_{Maj}	B_{Rep}	*Graph*	LC_d	LC_o
Acc	78.25	79.06	79.61	80.50	80.42
lit. Prec	-	70.00	52.21	62.26	53.89
lit. Rec	-	5.96	67.87	26.21	69.03
lit. $F_{\beta=1}$	-	**10.98**	**59.02**	**36.90**	**60.53**

- $\mathbf{B_{Maj}}$: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- $\mathbf{B_{Rep}}$: predict "literal" if an idiom component word is repeated in the context
- **Graph**: cohesion graph
- $\mathbf{LC_d}$: lexical chains optimised on development set
- $\mathbf{LC_o}$: lexical chains optimised globally by oracle (upper bound for lexical chains)

Results (More on Cohesion Graph)

Method	*Acc.*	LPrec.	LRec.	$LF_{\beta=1}$
Base	0.78	–	–	–
Base_r	0.50	0.22	0.51	0.30
Base_{r_con}	0.65	0.20	0.21	0.20
CGA	**0.79**	**0.50**	**0.69**	**0.58**
CGA_{para}	0.71	0.42	0.67	0.51
CGA_{prun}	0.78	0.49	0.72	0.58
CGA_{ew}	0.79	0.51	0.63	0.57
CGA_{nw}	0.77	0.48	0.68	0.56
CGA_{ew+nw}	0.78	0.49	0.61	0.54

- **Base**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **Base$_{\mathbf{r}}$**: random prediction
- **Base$_{\mathbf{r_con}}$**: random prediction with bias toward the non-literal class according to the true distribution
- **CGA**: cohesion graph

Results (More on Cohesion Graph)

Method	*Acc.*	LPrec.	LRec.	$LF_{\beta=1}$
Base	0.78	–	–	–
Base_r	0.50	0.22	0.51	0.30
Base_{r_con}	0.65	0.20	0.21	0.20
CGA	**0.79**	**0.50**	**0.69**	**0.58**
CGA_{para}	0.71	0.42	0.67	0.51
CGA_{prun}	0.78	0.49	0.72	0.58
CGA_{ew}	0.79	0.51	0.63	0.57
CGA_{nw}	0.77	0.48	0.68	0.56
CGA_{ew+nw}	0.78	0.49	0.61	0.54

- **Base**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **Base_r**: random prediction
- **$\text{Base}_\text{r\_con}$**: random prediction with bias toward the non-literal class according to the true distribution
- **CGA**: cohesion graph

Results (More on Cohesion Graph)

Method	*Acc.*	LPrec.	LRec.	$LF_{\beta=1}$
Base	0.78	–	–	–
Base_r	0.50	0.22	0.51	0.30
Base_{r_con}	0.65	0.20	0.21	0.20
CGA	**0.79**	**0.50**	**0.69**	**0.58**
CGA_{para}	0.71	0.42	0.67	0.51
CGA_{prun}	0.78	0.49	0.72	0.58
CGA_{ew}	0.79	0.51	0.63	0.57
CGA_{nw}	0.77	0.48	0.68	0.56
CGA_{ew+nw}	0.78	0.49	0.61	0.54

- **Base**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **Base$_\mathbf{r}$**: random prediction
- **Base$_\mathbf{r_con}$**: random prediction with bias toward the non-literal class according to the true distribution
- **CGA**: cohesion graph

Results (More on Cohesion Graph)

Method	*Acc.*	LPrec.	LRec.	$LF_{\beta=1}$
Base	0.78	–	–	–
Base_r	0.50	0.22	0.51	0.30
Base_{r_con}	0.65	0.20	0.21	0.20
CGA	**0.79**	**0.50**	**0.69**	**0.58**
CGA_{para}	0.71	0.42	0.67	0.51
CGA_{prun}	0.78	0.49	0.72	0.58
CGA_{ew}	0.79	0.51	0.63	0.57
CGA_{nw}	0.77	0.48	0.68	0.56
CGA_{ew+nw}	0.78	0.49	0.61	0.54

- **Base**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **$\mathbf{Base_r}$**: random prediction
- **$\mathbf{Base_{r\_con}}$**: random prediction with bias toward the non-literal class according to the true distribution
- **CGA**: cohesion graph

Results (More on Cohesion Graph)

Method	*Acc.*	LPrec.	LRec.	$LF_{\beta=1}$
Base	0.78	–	–	–
Base_r	0.50	0.22	0.51	0.30
Base_{r_con}	0.65	0.20	0.21	0.20
CGA	**0.79**	**0.50**	**0.69**	**0.58**
CGA_{para}	0.71	0.42	0.67	0.51
CGA_{prun}	0.78	0.49	0.72	0.58
CGA_{ew}	0.79	0.51	0.63	0.57
CGA_{nw}	0.77	0.48	0.68	0.56
CGA_{ew+nw}	0.78	0.49	0.61	0.54

- **Base**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **Base$_\mathbf{r}$**: random prediction
- **Base$_\mathbf{r_con}$**: random prediction with bias toward the non-literal class according to the true distribution
- **CGA**: cohesion graph

Results (More on Cohesion Graph)

Method	*Acc.*	LPrec.	LRec.	$LF_{\beta=1}$
Base	0.78	–	–	–
Base_r	0.50	0.22	0.51	0.30
Base_{r_con}	0.65	0.20	0.21	0.20
CGA	**0.79**	**0.50**	**0.69**	**0.58**
CGA_{para}	0.71	0.42	0.67	0.51
CGA_{prun}	0.78	0.49	0.72	0.58
CGA_{ew}	0.79	0.51	0.63	0.57
CGA_{nw}	0.77	0.48	0.68	0.56
CGA_{ew+nw}	0.78	0.49	0.61	0.54

- $\mathbf{CGA_{para}}$: cohesion graph built on the current paragraph
- $\mathbf{CGA_{prun}}$: pruning the 3 least connected nodes
- $\mathbf{CGA_{ew}}$: edge weights based on the inverse distance between the tokens
- $\mathbf{CGA_{nw}}$: node weights based on idf
- $\mathbf{CGA_{ew+nw}}$: edge weights plus node weights

Results (More on Cohesion Graph)

Method	*Acc.*	LPrec.	LRec.	$LF_{\beta=1}$
Base	0.78	–	–	–
Base_r	0.50	0.22	0.51	0.30
Base_{r_con}	0.65	0.20	0.21	0.20
CGA	**0.79**	**0.50**	**0.69**	**0.58**
CGA_{para}	0.71	0.42	0.67	0.51
CGA_{prun}	0.78	0.49	0.72	0.58
CGA_{ew}	0.79	0.51	0.63	0.57
CGA_{nw}	0.77	0.48	0.68	0.56
CGA_{ew+nw}	0.78	0.49	0.61	0.54

- $\mathbf{CGA_{para}}$: cohesion graph built on the current paragraph
- $\mathbf{CGA_{prun}}$: pruning the 3 least connected nodes
- $\mathbf{CGA_{ew}}$: edge weights based on the inverse distance between the tokens
- $\mathbf{CGA_{nw}}$: node weights based on idf
- $\mathbf{CGA_{ew+nw}}$: edge weights plus node weights

Results (More on Cohesion Graph)

Method	*Acc.*	LPrec.	LRec.	$LF_{\beta=1}$
Base	0.78	–	–	–
Base_r	0.50	0.22	0.51	0.30
Base_{r_con}	0.65	0.20	0.21	0.20
CGA	**0.79**	**0.50**	**0.69**	**0.58**
CGA_{para}	0.71	0.42	0.67	0.51
CGA_{prun}	0.78	0.49	0.72	0.58
CGA_{ew}	0.79	0.51	0.63	0.57
CGA_{nw}	0.77	0.48	0.68	0.56
CGA_{ew+nw}	0.78	0.49	0.61	0.54

- $\mathbf{CGA_{para}}$: cohesion graph built on the current paragraph
- $\mathbf{CGA_{prun}}$: pruning the 3 least connected nodes
- $\mathbf{CGA_{ew}}$: edge weights based on the inverse distance between the tokens
- $\mathbf{CGA_{nw}}$: node weights based on idf
- $\mathbf{CGA_{ew+nw}}$: edge weights plus node weights

Results (More on Cohesion Graph)

Method	*Acc.*	LPrec.	LRec.	$LF_{\beta=1}$
Base	0.78	–	–	–
$Base_r$	0.50	0.22	0.51	0.30
$Base_{r\_con}$	0.65	0.20	0.21	0.20
CGA	**0.79**	**0.50**	**0.69**	**0.58**
CGA_{para}	0.71	0.42	0.67	0.51
CGA_{prun}	0.78	0.49	0.72	0.58
CGA_{ew}	0.79	0.51	0.63	0.57
CGA_{nw}	0.77	0.48	0.68	0.56
CGA_{ew+nw}	0.78	0.49	0.61	0.54

- **CGA_{para}**: cohesion graph built on the current paragraph
- **CGA_{prun}**: pruning the 3 least connected nodes
- **CGA_{ew}**: edge weights based on the inverse distance between the tokens
- **CGA_{nw}**: node weights based on idf
- **CGA_{ew+nw}**: edge weights plus node weights

Results (Supervised Classifier)

Feature	Avg. literal (%) Prec.	Rec.	F-Score	Avg. (%) Acc.
salW	77.10	56.10	65.00	86.83
relW	78.00	43.20	55.60	84.99
relS	74.90	37.50	50.00	83.68
connectivity	78.30	2.10	4.10	78.58
salW+relW+relS	82.90	63.50	71.90	89.20
all	85.80	66.60	75.00	90.34

Results (Supervised Classifier)

Feature	Avg. literal (%) Prec.	Rec.	F-Score	Avg. (%) Acc.
salW	77.10	56.10	65.00	86.83
relW	78.00	43.20	55.60	84.99
relS	74.90	37.50	50.00	83.68
connectivity	78.30	2.10	4.10	78.58
salW+relW+relS	82.90	63.50	71.90	89.20
all	85.80	66.60	75.00	90.34

- The *salient words* feature has the highest performance

Results (Supervised Classifier)

Feature	Avg. literal (%) Prec.	Rec.	F-Score	Avg. (%) Acc.
salW	77.10	56.10	65.00	86.83
relW	78.00	43.20	55.60	84.99
relS	74.90	37.50	50.00	83.68
connectivity	78.30	2.10	4.10	78.58
salW+relW+relS	82.90	63.50	71.90	89.20
all	85.80	66.60	75.00	90.34

- The *salient words* feature has the highest performance
- *Connectivity gain* feature increases the performance of the model combined with the other features

Results (Combined Classifier)

Model	Acc.	$Prec_l$	Rec_l	F-$Score_l$
$Base_{maj}$	78.25	-	-	-
unsup.	78.38	50.04	69.72	58.26
combined	86.30	83.86	45.82	59.26
combined+boost	86.13	70.26	62.76	66.30
combined+it*	86.68	85.68	46.52	60.30
combined+boost+it*	**87.03**	**71.86**	**66.36**	**69.00**
super. 10CV	90.34	85.80	66.60	75.00

- **$\mathbf{Base_{maj}}$**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **unsup**.: cohesion graph (Sporleder and Li (2009))
- **combined**: combined classifier
- **combined** + **boost**: combined classifier with boosting literal class
- **combined** + +**boost** + **it***: iteratively increase trainning set, boosting literal class in each iteration
- **super.10CV**: 10-fold cross validation for supervised classifier

Results (Combined Classifier)

Model	Acc.	$Prec_l$	Rec_l	F-$Score_l$
$Base_{maj}$	78.25	-	-	-
unsup.	78.38	50.04	69.72	58.26
combined	86.30	83.86	45.82	59.26
combined+boost	86.13	70.26	62.76	66.30
combined+it*	86.68	85.68	46.52	60.30
combined+boost+it*	**87.03**	**71.86**	**66.36**	**69.00**
super. 10CV	90.34	85.80	66.60	75.00

- **$Base_{maj}$**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **unsup.**: cohesion graph (Sporleder and Li (2009))
- **combined**: combined classifier
- **combined + boost**: combined classifier with boosting literal class
- **combined + +boost + it***: iteratively increase trainning set, boosting literal class in each iteration
- **super.10CV**: 10-fold cross validation for supervised classifier

Results (Combined Classifier)

Model	Acc.	$Prec_l$	Rec_l	$F\text{-}Score_l$
$Base_{maj}$	78.25	-	-	-
unsup.	78.38	50.04	69.72	58.26
combined	86.30	83.86	45.82	59.26
combined+boost	86.13	70.26	62.76	66.30
combined+it*	86.68	85.68	46.52	60.30
combined+boost+it*	**87.03**	**71.86**	**66.36**	**69.00**
super. 10CV	90.34	85.80	66.60	75.00

- **$\mathbf{Base_{maj}}$**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **unsup**.: cohesion graph (Sporleder and Li (2009))
- **combined**: combined classifier
- **combined** + **boost**: combined classifier with boosting literal class
- **combined** + +**boost** + **it***: iteratively increase trainning set, boosting literal class in each iteration
- **super.10CV**: 10-fold cross validation for supervised classifier

Results (Combined Classifier)

Model	Acc.	$Prec_l$	Rec_l	$F\text{-}Score_l$
$Base_{maj}$	78.25	-	-	-
unsup.	78.38	50.04	69.72	58.26
combined	86.30	83.86	45.82	59.26
combined+boost	86.13	70.26	62.76	66.30
combined+it*	86.68	85.68	46.52	60.30
combined+boost+it*	**87.03**	**71.86**	**66.36**	**69.00**
super. 10CV	90.34	85.80	66.60	75.00

- **$\mathbf{Base_{maj}}$**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **unsup.**: cohesion graph (Sporleder and Li (2009))
- **combined**: combined classifier
- **combined + boost**: combined classifier with boosting literal class
- **combined + +boost + it***: iteratively increase trainning set, boosting literal class in each iteration
- **super.10CV**: 10-fold cross validation for supervised classifier

Results (Combined Classifier)

Model	Acc.	$Prec_l$	Rec_l	$F\text{-}Score_l$
$Base_{maj}$	78.25	-	-	-
unsup.	78.38	50.04	69.72	58.26
combined	86.30	83.86	45.82	59.26
combined+boost	86.13	70.26	62.76	66.30
combined+it*	86.68	85.68	46.52	60.30
combined+boost+it*	**87.03**	**71.86**	**66.36**	**69.00**
super. 10CV	90.34	85.80	66.60	75.00

- **$Base_{maj}$**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **unsup**.: cohesion graph (Sporleder and Li (2009))
- **combined**: combined classifier
- **combined + boost**: combined classifier with boosting literal class
- **combined + +boost + it***: iteratively increase trainning set, boosting literal class in each iteration
- **super.10CV**: 10-fold cross validation for supervised classifier

Results (Combined Classifier)

Model	Acc.	$Prec_l$	Rec_l	F-$Score_l$
$Base_{maj}$	78.25	-	-	-
unsup.	78.38	50.04	69.72	58.26
combined	86.30	83.86	45.82	59.26
combined+boost	86.13	70.26	62.76	66.30
combined+it*	86.68	85.68	46.52	60.30
combined+boost+it*	**87.03**	**71.86**	**66.36**	**69.00**
super. 10CV	90.34	85.80	66.60	75.00

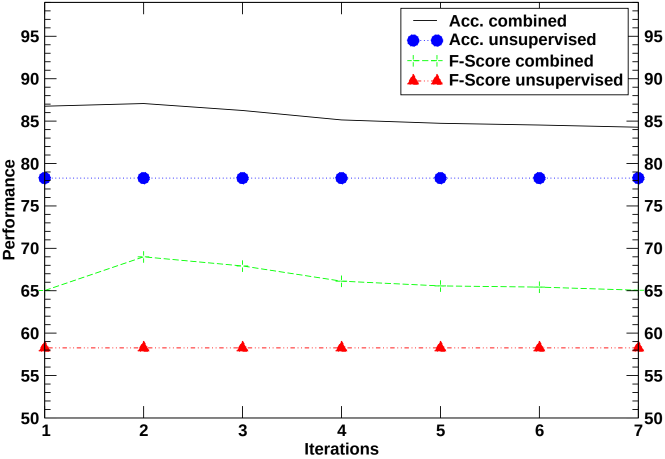
- **$Base_{maj}$**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **unsup**.: cohesion graph (Sporleder and Li (2009))
- **combined**: combined classifier
- **combined** + **boost**: combined classifier with boosting literal class
- **combined** + +**boost** + **it***: iteratively increase trainning set, boosting literal class in each iteration
- **super.10CV**: 10-fold cross validation for supervised classifier

Results (Combined Classifier)

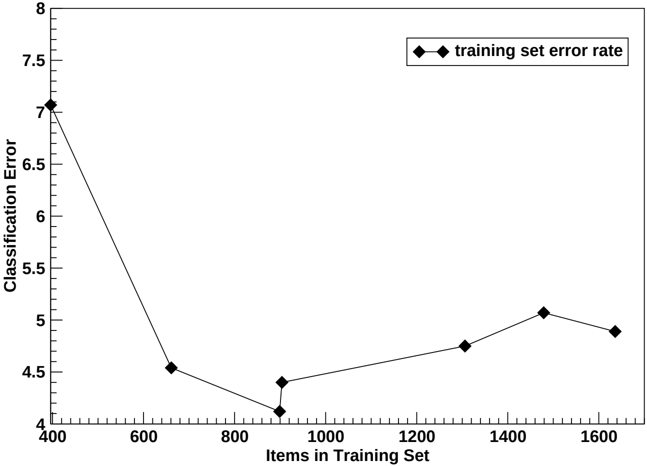
Model	Acc.	$Prec_l$	Rec_l	F-$Score_l$
$Base_{maj}$	78.25	-	-	-
unsup.	78.38	50.04	69.72	58.26
combined	86.30	83.86	45.82	59.26
combined+boost	86.13	70.26	62.76	66.30
combined+it*	86.68	85.68	46.52	60.30
combined+boost+it*	**87.03**	**71.86**	**66.36**	**69.00**
super. 10CV	90.34	85.80	66.60	75.00

- **$Base_{maj}$**: majority baseline, i.e., "non-literal" (cf. CForm classifier by Cook et al. (2007), Fazly et al. (to appear))
- **unsup**.: cohesion graph (Sporleder and Li (2009))
- **combined**: combined classifier
- **combined** + **boost**: combined classifier with boosting literal class
- **combined** + +**boost** + **it***: iteratively increase trainning set, boosting literal class in each iteration
- **super.10CV**: 10-fold cross validation for supervised classifier

Iterative Training with Boosting Literal Class



Error in the Training Set



Conclusion (1)

Lexical Chain & Cohesion Graph:

- The cohesive structure of a text provides good cues for distinguishing literal and non-literal language.
- Cohesion graphs perform as well as lexical chains while being fully unsupervised
- Varieties of the Cohesion Graph approach (weighted version and pruning experiments) do not work well

Conclusion (2)

More Sophisticated Classification Models:

- Experiment with linguistically informed features for a supervised classifier
- Build a bootstrapping model that complements an unsupervised classifier with a supervised classifier. This model explores lexical cohesion features and other linguistic clues
- The combined classifier can lead to significant reduction of classification errors
- Performance can be improved further by boosting the literal cases, which can be automatically extracted from an unlabeled corpus

References

- Caroline Sporleder and Linlin Li. Unsupervised Recognition of Literal and Non-Literal Use of Idiomatic Expressions. In *Proceedings of 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, April 1-3, 2009. Athens, Greece
- Linlin Li and Caroline Sporleder. Classifier combination for Contextual idiom detection without labelled data. In *Proceeding of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, August 6-7, 2009. Singapore
- Linlin Li and Caroline Sporleder. A cohesion graph based approach for unsupervised recognition of literal and nonliteral use of multiword expressions. In *Proceedings of the ACL 2009 Workshop on TextGraphs-4: Graph-based Methods for Natural Language Processing*, August 7, 2009. Singapore
- Linlin Li, and Caroline Sporleder. Using Gaussian Mixture Models to Detect Figurative Language in Context. In *Proceedings of the 11th Annual Conference of North American Chapter of Association for Computational Linguistics (NAACL-HLT)* , Short Papers. June 1-6, 2010. LA, US