Predicting Thread Linking Structure by Lexical Chaining

Li Wang,[♠♡] Diana McCarthy[♢] and Timothy Baldwin[♠♡]

♠ Dept. of Computer Science and Software Engineering, University of Melbourne
♡ NICTA Victoria Research Laboratory
♢ Lexical Computing Ltd

December 2nd, 2011

Introduction

Example Thread

HTML Input Code - CNET Coding & scripting Forums

User A Post 1	**HTML Input Code** ...Please can someone tell me how to create an input box that asks the user to enter their ID, and then allows them to press go. It will then redirect to the page ...
User B Post 2	**Re: html input code** Part 1: create a form with a text field. See ... Part 2: give it a Javascript action ...
User C Post 3	**asp.net c\# video** I've prepared for you video.link click ...
User A Post 4	**Thank You!** Thanks a lot for that ... I have Microsoft Visual Studio 6, what program should I do this in? Lastly, how do I actually include this in my site? ...
User D Post 5	**A little more help** ... You would simply do it this way: ... You could also just ... An example of this is ...

Resources: http://forums.cnet.com/

Example Thread

HTML Input Code - CNET Coding & scripting Forums

User A Post 1	**HTML Input Code** ...Please can someone tell me how to create an input box that asks the user to enter their ID, and then allows them to press go. It will then redirect to the page ...
User B Post 2	**Re: html input code** Part 1: create a form with a text field. See ... Part 2: give it a Javascript action ...
User C Post 3	**asp.net c\# video** I've prepared for you video.link click ...
User A Post 4	**Thank You!** Thanks a lot for that ... I have Microsoft Visual Studio 6, what program should I do this in? Lastly, how do I actually include this in my site? ...
User D Post 5	**A little more help** ... You would simply do it this way: ... You could also just ... An example of this is ...

External Link

External Video

500 words in total

Resources: http://forums.cnet.com/

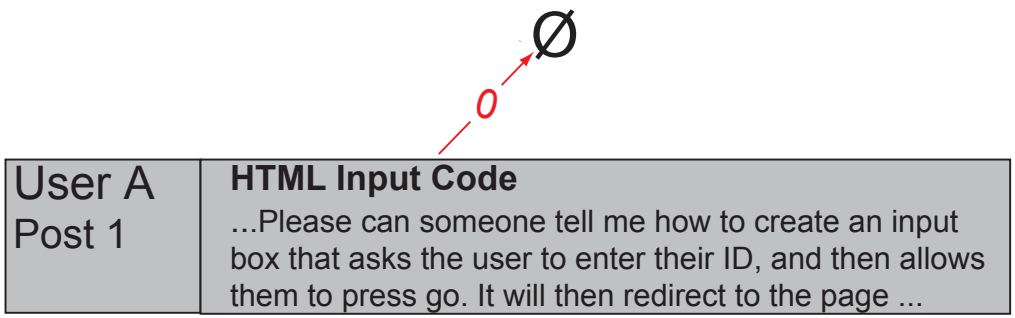
Research Motivation and Aim

- Motivation:
  - thread linking structure can be used to improve information retrieval in forums.
  - linking information is not supported in many forums.
  - most research focuses on supervised methods.
- Aim:
  - recover forum thread linking structure by lexical chaining.

Linking Structure of Forum Threads

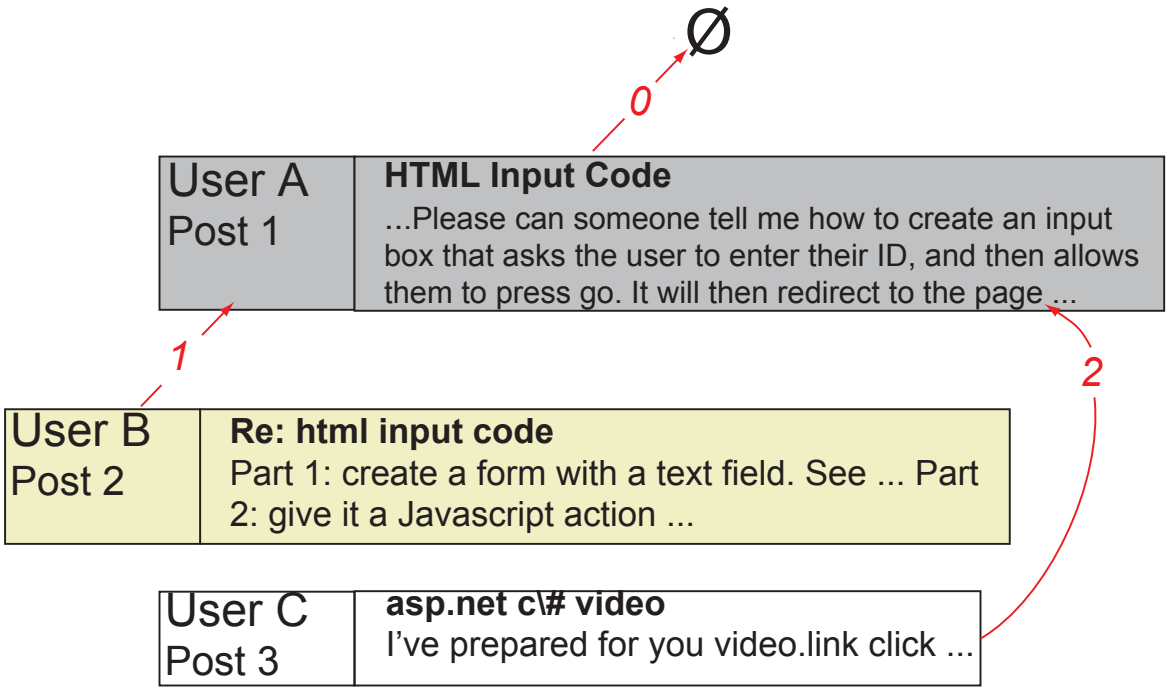
Reference: Kim et al., 2010

Linking Structure of Forum Threads



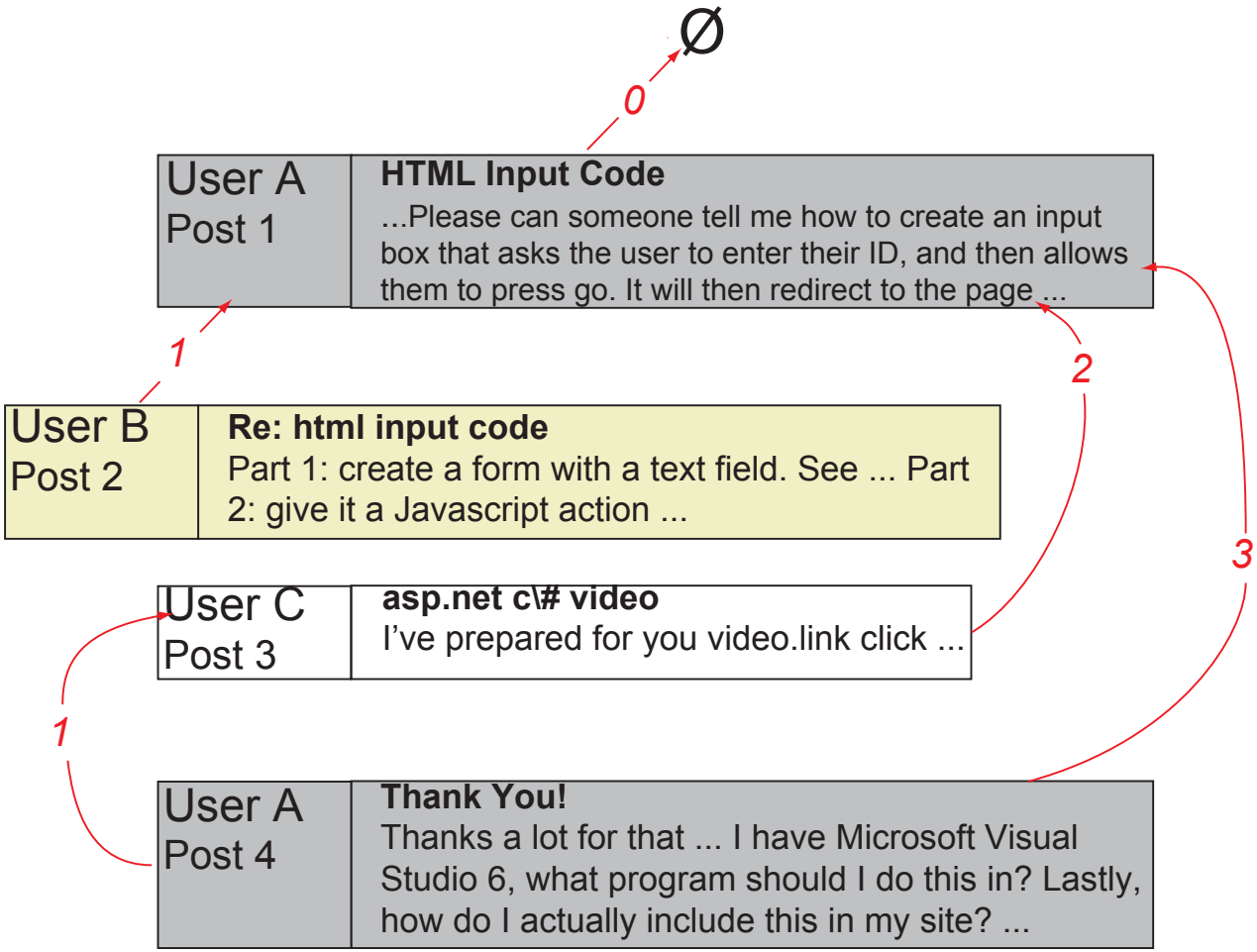
Reference: Kim et al., 2010

Linking Structure of Forum Threads



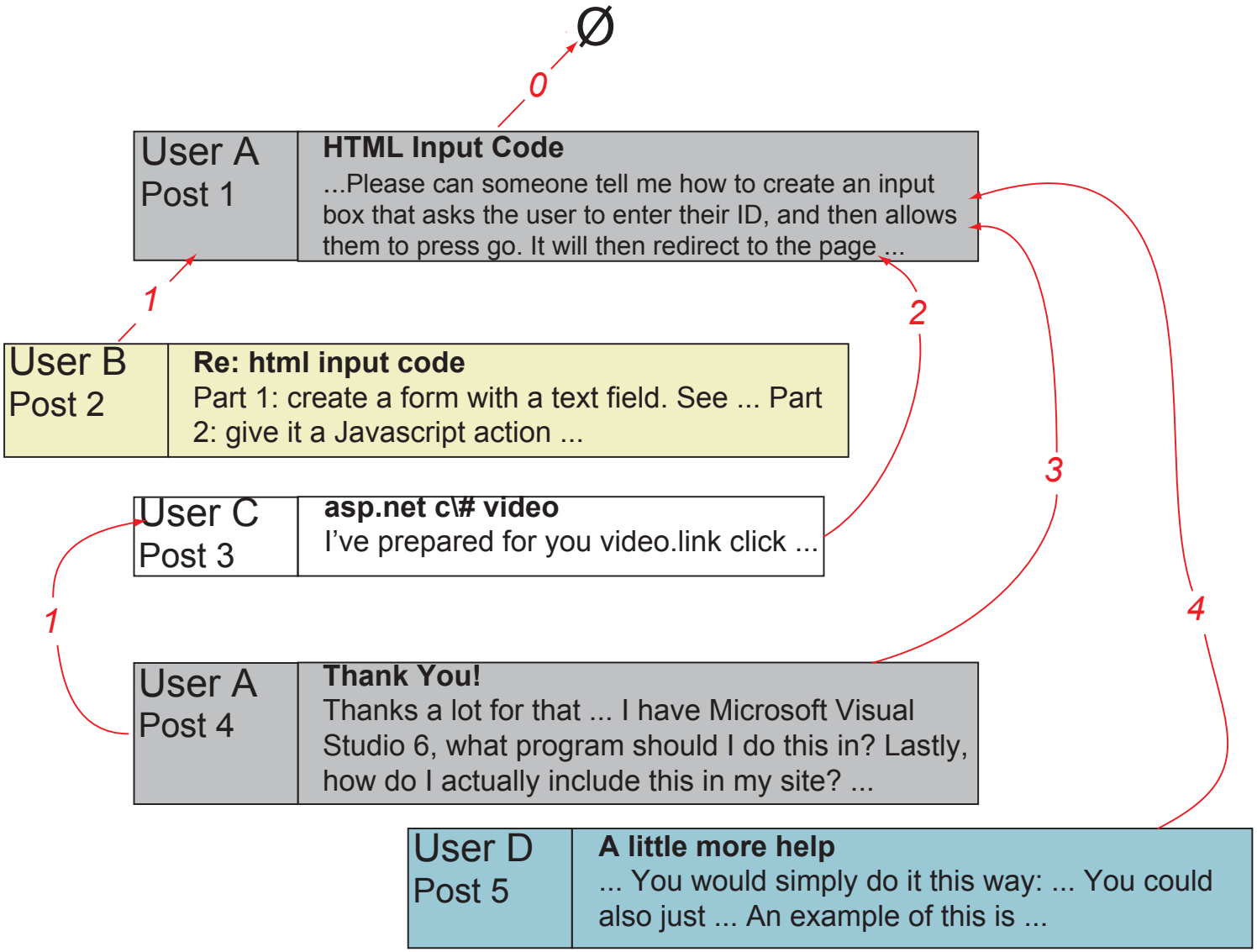
Reference: Kim et al., 2010

Linking Structure of Forum Threads



Reference: Kim et al., 2010

Linking Structure of Forum Threads



Reference: Kim et al., 2010

Lexical Chaining

- Lexical chaining is a technique for identifying lists of related words (lexical chains) within a given discourse.
- An Example:

User A Post 1	**HTML Input Code** ...I am new to HTML Coding. Please can someone tell me how to create an input box that asks the user to enter their ID, and then allows them to press go. It will then redirect to the page ... For example, if they input an ID of 12345 ...
User B Post 2	**Re: html input code** Part 1: create a form with a text field. See ... Part 2: give it a Javascript action that composes the full address of the new page and jumps to it. But as you don't ask for Javascript coding, I won't elaborate on that ...
User C Post 3	**asp.net c\# video** I've prepared for you video.link click ...

Lexical Chaining

- Lexical chaining is a technique for identifying lists of related words (lexical chains) within a given discourse.
- An Example:

User A Post 1	**HTML Input Code** ...I am new to HTML Coding. Please can someone tell me how to create an input box that asks the user to enter their ID, and then allows them to press go. It will then redirect to the page ... For example, if they input an ID of 12345 ...
User B Post 2	**Re: html input code** Part 1: create a form with a text field. See ... Part 2: give it a Javascript action that composes the full address of the new page and jumps to it. But as you don't ask for Javascript coding, I won't elaborate on that ...

Lexical Chaining

- Lexical chaining is a technique for identifying lists of related words (lexical chains) within a given discourse.
- An Example:

User A Post 1	**HTML Input Code** ...I am new to HTML Coding. Please can someone tell me how to create an input box that asks the user to enter their ID, and then allows them to press go. It will then redirect to the page ... For example, if they input an ID of 12345 ...

User C Post 3	**asp.net c\# video** I've prepared for you video.link click ...

Lexical Chaining

- Lexical chaining is a technique for identifying lists of related words (lexical chains) within a given discourse.
- An Example:

User B Post 2	**Re: html input code** Part 1: create a form with a text field. See ... Part 2: give it a Javascript action that composes the full address of the new page and jumps to it. But as you don't ask for Javascript coding, I won't elaborate on that ...
User C Post 3	**asp.net c\# video** I've prepared for you video.link click ...

Lexical Chaining Algorithms

Chainer$_{Roget}$

- A Roget's Thesaurus based lexical chaining algorithm from the Electronic Lexical Knowledge Base (ELKB).
- Key steps:
  - build all possible chains for each candidate noun separately. Each pair of nouns in each chain are either the same word or included in the same *Head* of *Roget's Thesaurus*.
  - select the strongest chain for each candidate noun.
  - merge two chains if they contain at least one noun in common

Reference: Jarmasz and Szpakowicz, 2001, 2003

$Chainer_{Roget}$ . . .

- Example:

w1	w2	w3	w4	w3	w5
w6	w5	w7	w8	w9	w10

w3	w11	w12	w9

$Chainer_{Roget}$ . . .

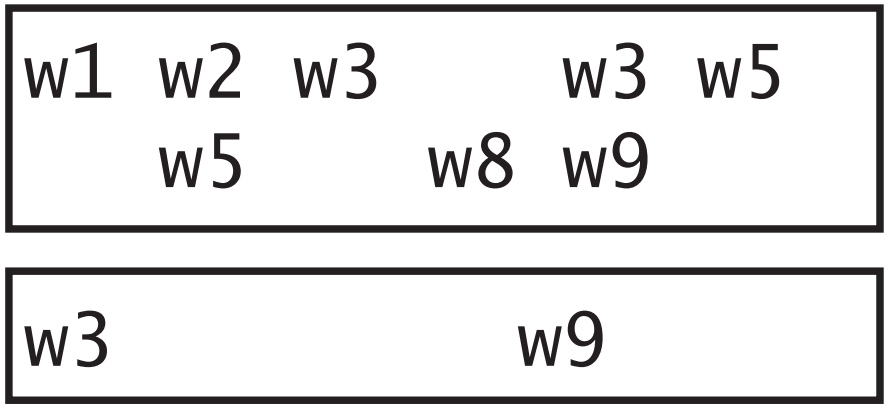
- Example:

w1	w2	w3			w3	w5	
	w5			w8	w9		

w3					w9		

$Chainer_{Roget}$ . . .

- Example:




```
Head1: {w1, w3, w5}
Head2: {w3, w8, w9}
```

Chainer$_{Roget}$ ...

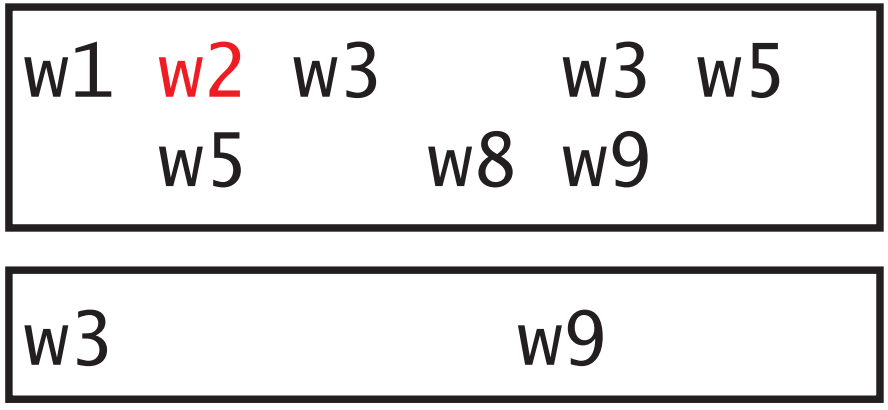
- Example:

Chain1:
[w1,w3,w3,w3,w5,w5]

Head1: {w1, w3, w5}
Head2: {w3, w8, w9}

Chainer$_{Roget}$ ...

- Example:

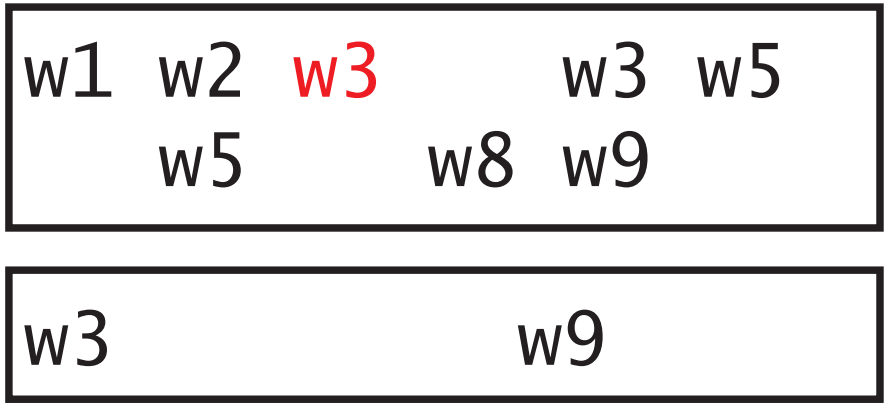


Chain1:
[w1,w3,w3,w3,w5,w5]

Head1: {w1, w3, w5}
Head2: {w3, w8, w9}

Chainer$_{Roget}$ ...

- Example:



Head1: {w1, w3, w5}
Head2: {w3, w8, w9}

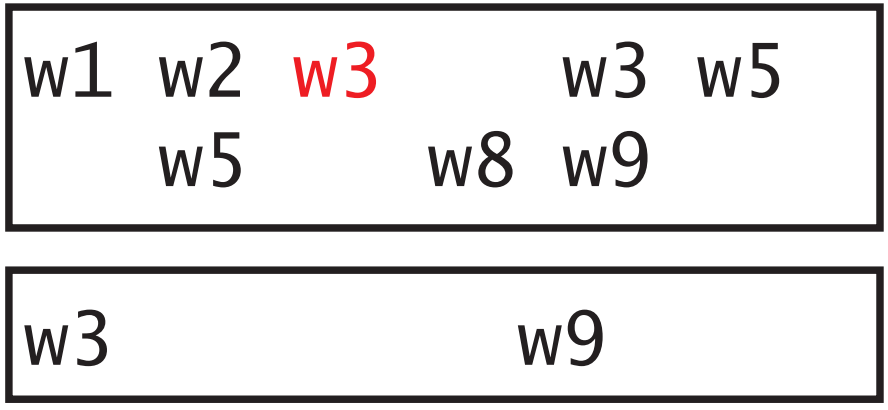
Chain1:
[w1,w3,w3,w3,w5,w5]

Chain2:
[w3,w3,w3,w5,w5]

Chain3:
[w3,w3,w8,w9,w3]

Chainer$_{Roget}$ ...

- Example:



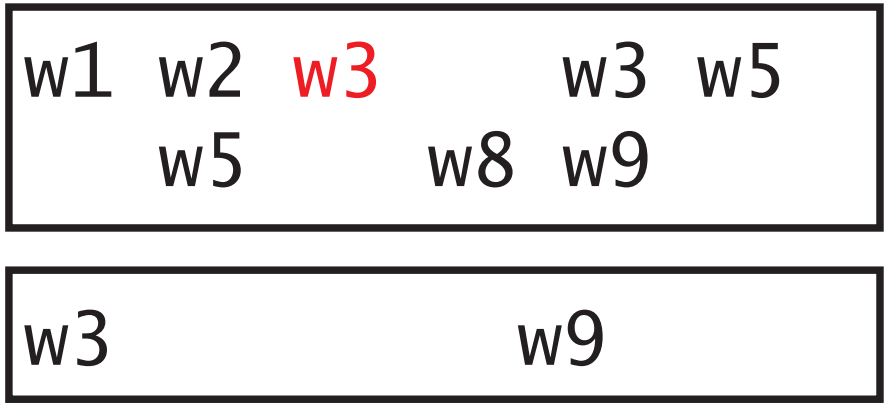
Chain1:
[w1,w3,w3,w3,w5,w5]

Chain2:
[w3,w3,w3,w5,w5]

Head1: {w1, w3, w5}
Head2: {w3, w8, w9}

Chainer$_{Roget}$ ...

- Example:



Chain4:
[w1,w3,w3,w3,w5,w5]

Head1: {w1, w3, w5}
Head2: {w3, w8, w9}

$Chainer_{WN}$

- A non-greedy WordNet-based chaining algorithm proposed by Galley and McKeown [2003].
- Key steps:
  - build a Disambiguation Graph: each node represents a word, and each edge connects two senses of two words.
  - perform Word Sense Disambiguation: the word sense with the highest sum of edge weights (one sense per word).
  - generate lexical chains: remove edges connecting "wrong" senses.

$Chainer_{WN}$ . . .

- Example:

```
w1 w2 w3 w4 w3 w5
w6 w5 w7 w8 w9 w10
```

```
w3 w11 w12 w9
```

$Chainer_{WN}$ . . .

- Example:

w1	w2	w3		w3	w5
	w5		w8	w9	

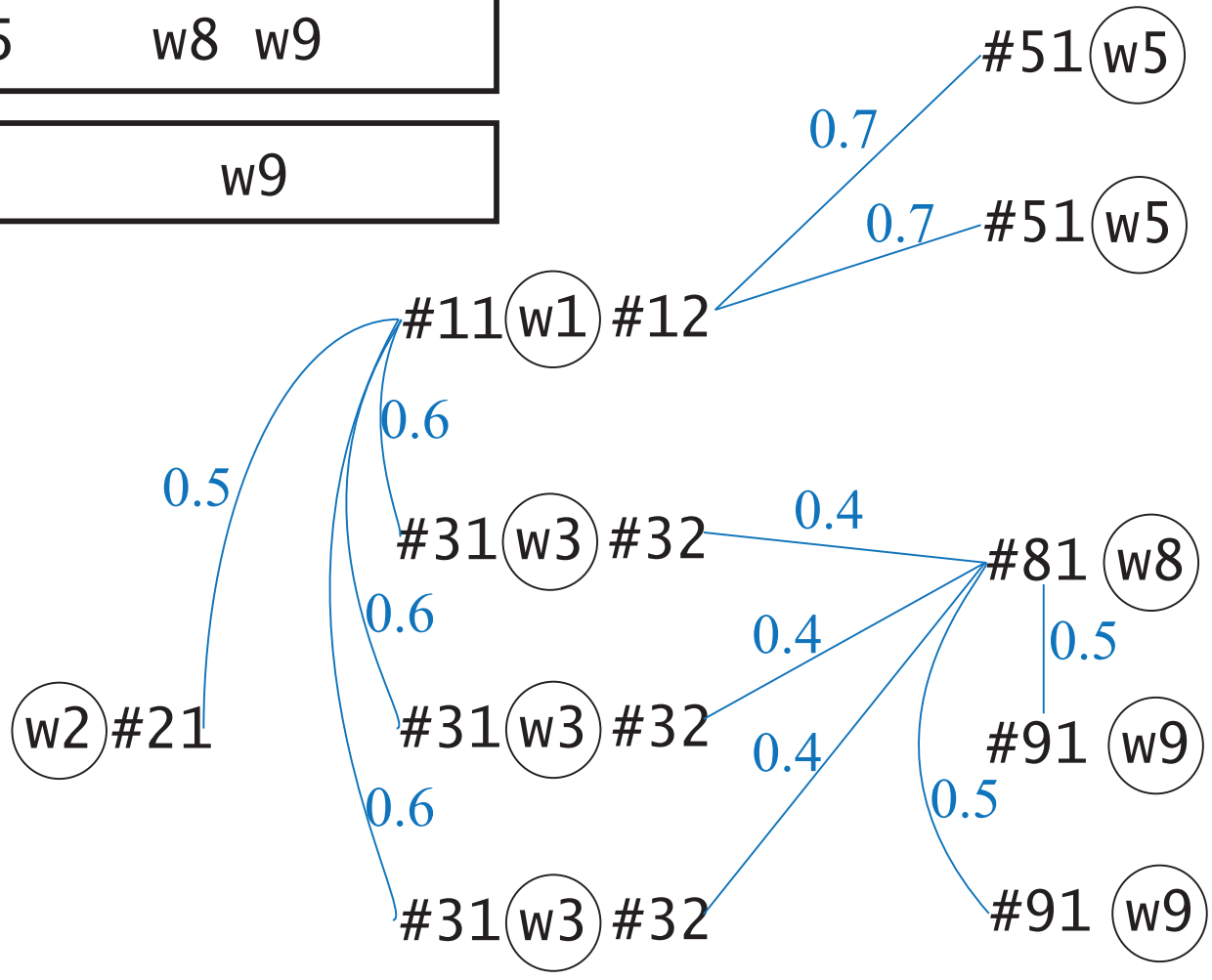
w3		w9

$Chainer_{WN}$ . . .

- Example:

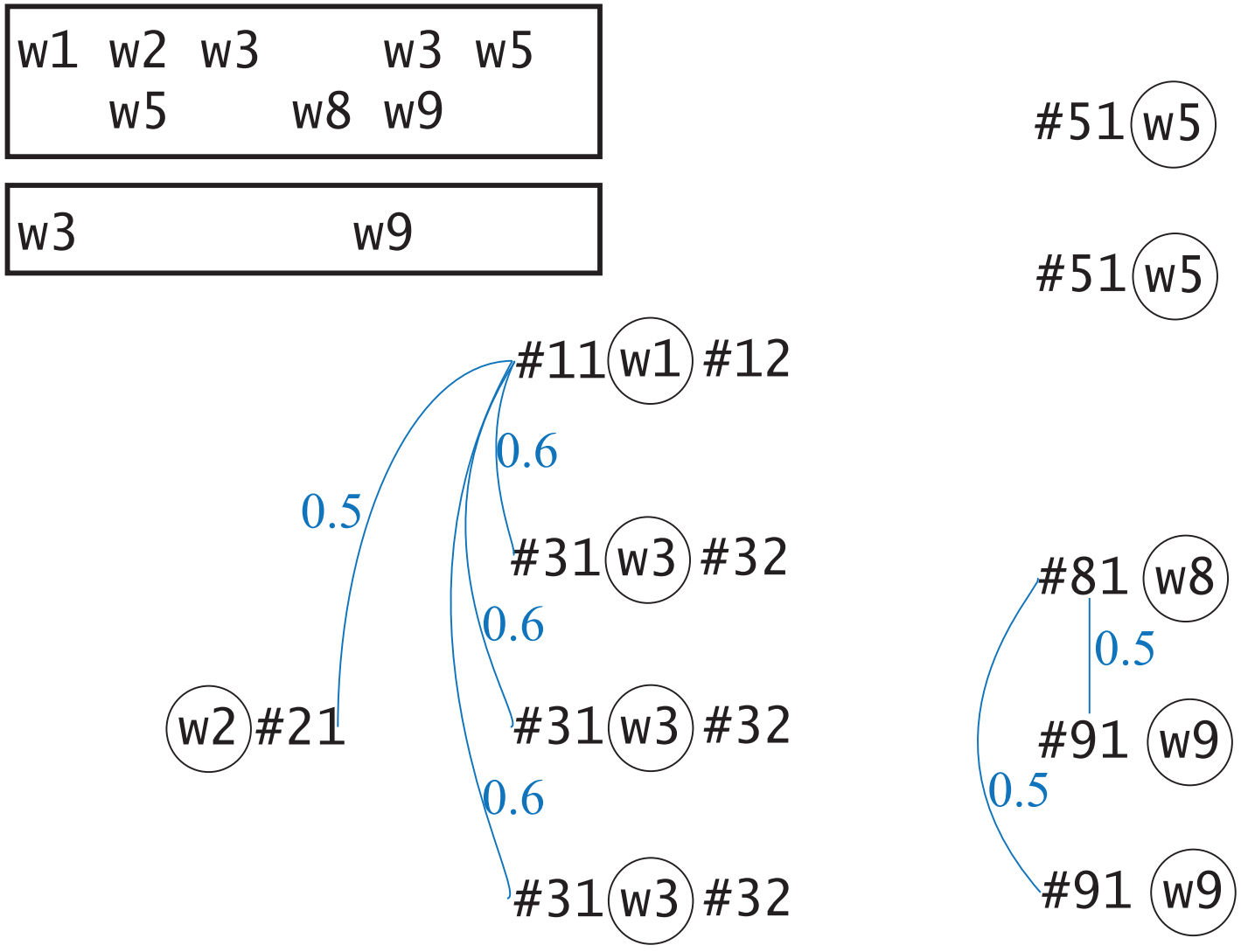
w1	w2	w3		w3	w5
	w5		w8	w9	

w3		w9



$Chainer_{WN}$ . . .

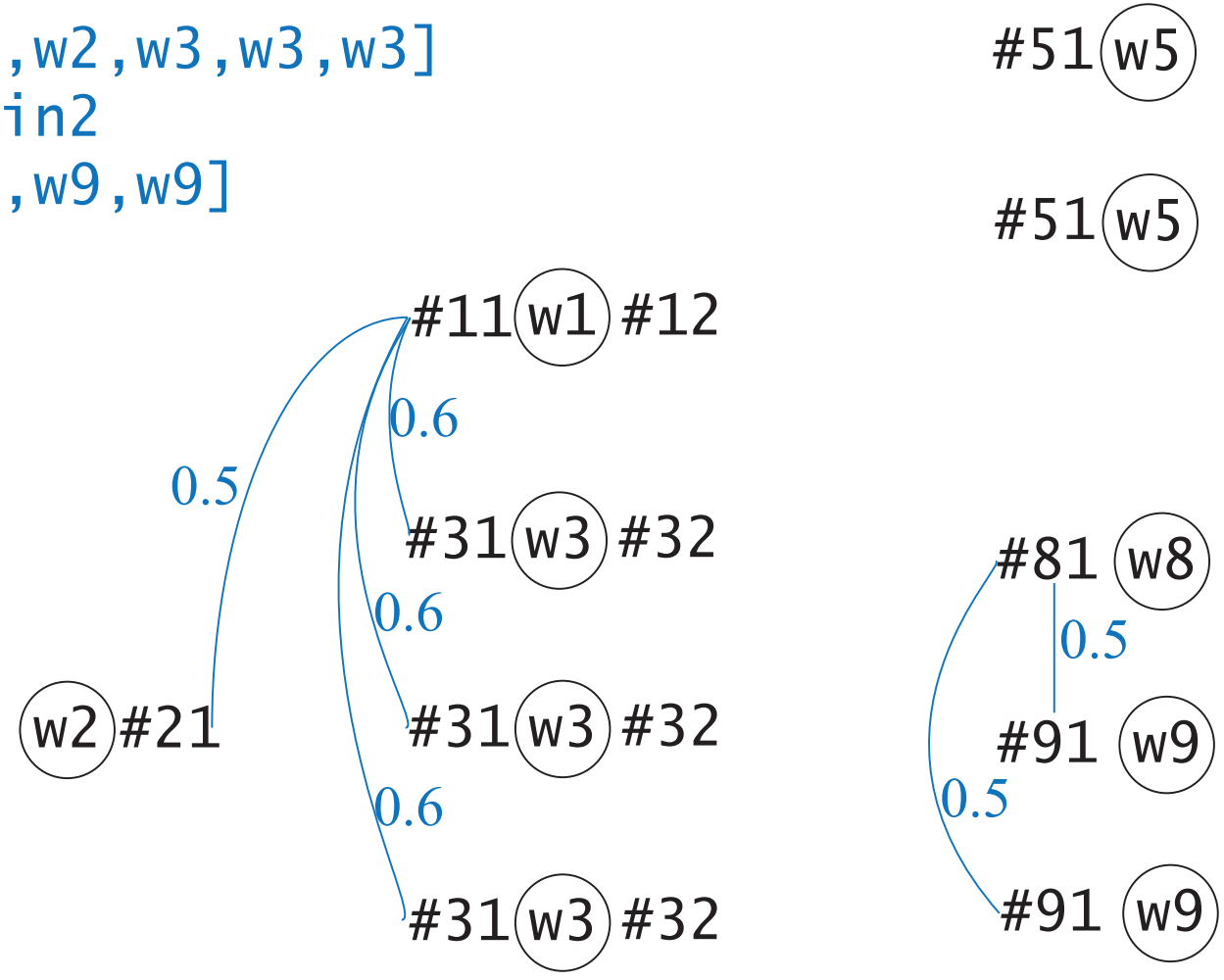
- Example:



$Chainer_{WN}$ . . .

- Example:

```
Chain1
[w1,w2,w3,w3,w3]
Chain2
[w8,w9,w9]
```



$Chainer_{SV}$

- A statistical association based chaining algorithm adapted from Marathe and Hirst [2010].
- Key steps:
  - calculate word-word similarity based on word vectors, which are from a WORDSPACE model built with SemanticVectors.
  - add each word to the chain with the largest word-chain similarity.
  - if there are multiple candidate chains, merge chains based on chain-chain similarities.

Reference: Schütze, 1998, Widdows and Ferraro, 2008

$Chainer_{SV}$ . . .

- Example:

```
w1 w2 w3 w4 w3 w5
w6 w5 w7 w8 w9 w10
```

```
w3 w11 w12 w9
```

$Chainer_{SV}$ . . .

- Example:

w1	w2	w3		w3	w5
	w5		w8	w9	

w3		w9

$Chainer_{SV}$ ...

- Example:

Sim(word,word)
Sim(chain,word)
Sim(chain,chain)

Threshold-add
Threshold-merge

$Chainer_{SV}$ ...

- Example:

w1	w2	w3			w3	w5	
	w5			w8	w9		

w3		w9

Chains:
[w1]

Sim(word,word)
Sim(chain,word)
Sim(chain,chain)

Threshold-add
Threshold-merge

$Chainer_{SV}$ ...

- Example:

Chains:
[w1]

Sim(word,word)
Sim(chain,word)
Sim(chain,chain)

Threshold-add
Threshold-merge

$Chainer_{SV}$ ...

- Example:

Chains:
[w1]
[w2]

Sim(word,word)
Sim(chain,word)
Sim(chain,chain)

Threshold-add
Threshold-merge

$Chainer_{SV}$ ...

- Example:

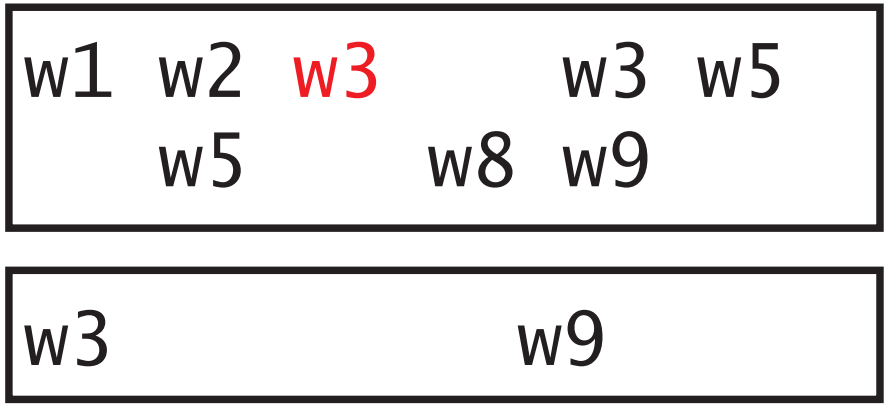
w1 w2 w3 w3 w5 w5 w8 w9	Chains: [w1] [w2]
w3 w9	

Sim(word,word)
Sim(chain,word)
Sim(chain,chain)

Threshold-add
Threshold-merge

$Chainer_{SV}$ ...

- Example:



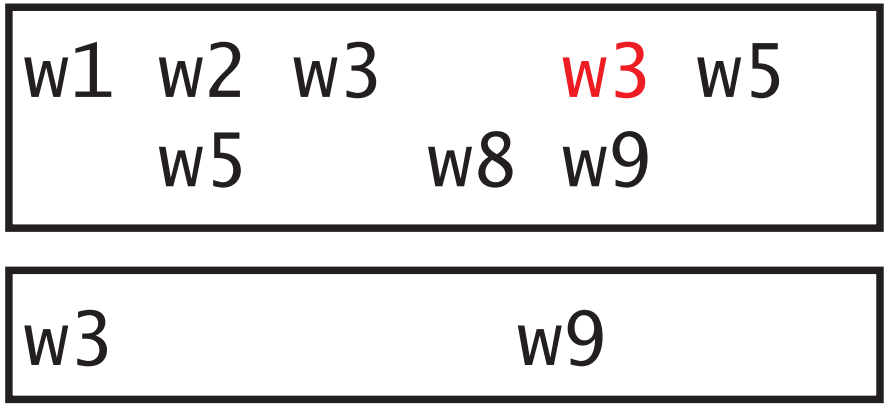
Chains:
[w1,w3]
[w2]

Sim(word,word)
Sim(chain,word)
Sim(chain,chain)

Threshold-add
Threshold-merge

$Chainer_{SV}$ ...

- Example:



Chains:
[w1,w3,w3]
[w2]

Sim(word,word)
Sim(chain,word)
Sim(chain,chain)

Threshold-add
Threshold-merge

$Chainer_{SV}$ ...

- Example:

Chains:
[w1,w3,w3]
[w2]

Sim(word,word)
Sim(chain,word)
Sim(chain,chain)

Threshold-add
Threshold-merge

$Chainer_{SV}$ ...

- Example:

Chains:
[w1,w3,w3]
[w2]

Sim(word,word)
Sim(chain,word)
Sim(chain,chain)

Threshold-add
Threshold-merge

$Chainer_{SV}$ ...

- Example:

Chains:
[w1,w3,w3,w5]
[w2]

Sim(word,word)
Sim(chain,word)
Sim(chain,chain)

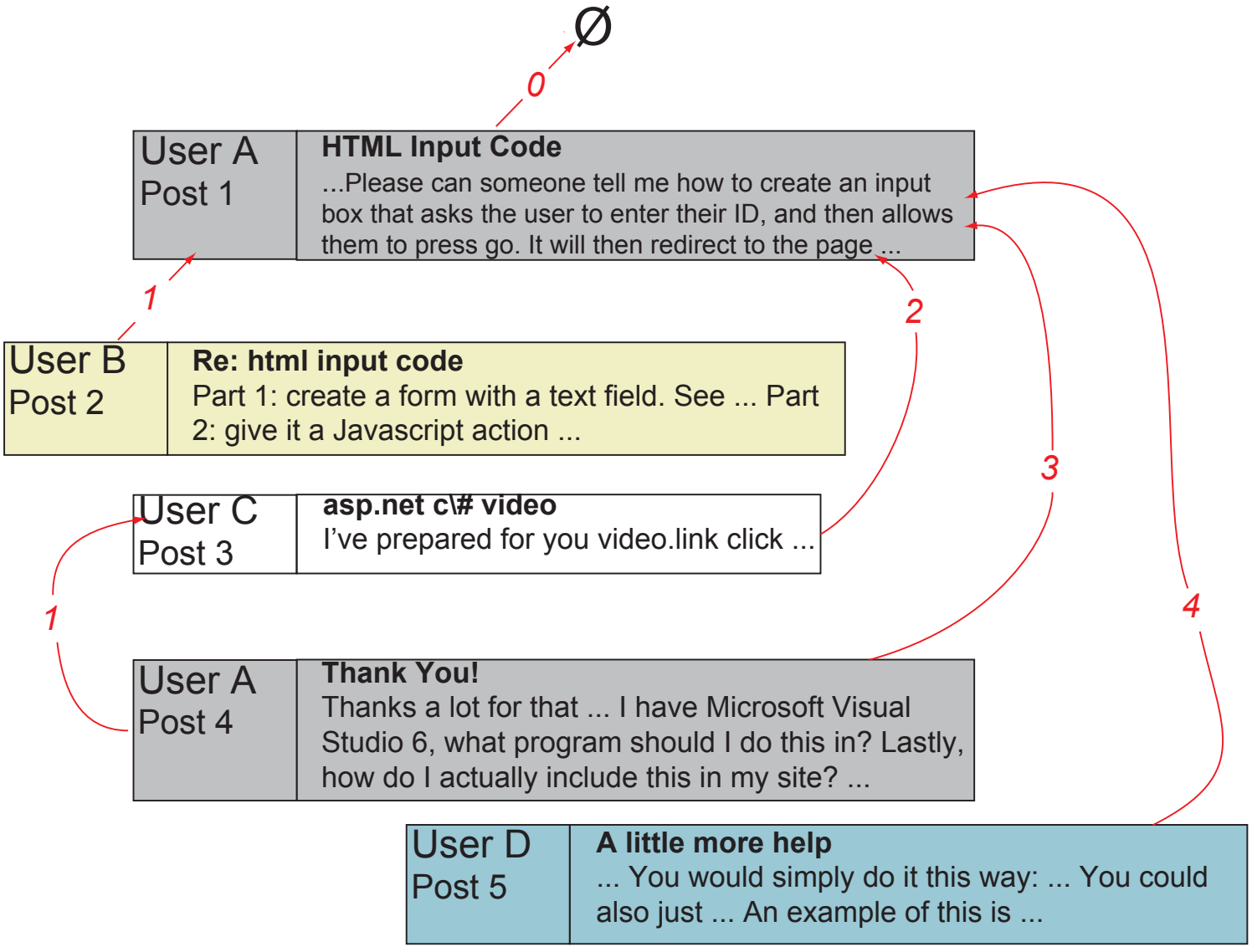
Threshold-add
Threshold-merge

Experimental Setup

Datasets

- Dataset with annotation:
  - from Kim et al. [2010], 1332 posts spanning 315 threads from the Operating System, Software, Hardware and Web Development sub-forums of CNET.
  - each post is labelled with one or more links, each link is labelled with a dialogue act. Only links are used in this research.
- Dataset without annotation:
  - used for building the WORDSPACE model.
  - 262,402 threads in total from CNET.
  - only the threads from the four sub-forums are chosen, which consist of 536,482 posts spanning 114,139 threads.

Recap


Ø
0
User A
Post 1
HTML Input Code
...Please can someone tell me how to create an input box that asks the user to enter their ID, and then allows them to press go. It will then redirect to the page ...
1
User B
Post 2
Re: html input code
Part 1: create a form with a text field. See ... Part 2: give it a Javascript action ...
2
User C
Post 3
asp.net c\# video
I've prepared for you video.link click ...
3
1
User A
Post 4
Thank You!
Thanks a lot for that ... I have Microsoft Visual Studio 6, what program should I do this in? Lastly, how do I actually include this in my site? ...
4
User D
Post 5
A little more help
... You would simply do it this way: ... You could also just ... An example of this is ...

Basic Idea

- Basic idea: use lexical chains to measure the two posts' lexical cohesion (i.e. lexical similarity):
  - combine the texts from the titles and bodies of the two posts.
  - extract lexical chains from the combined texts
  - calculate the lexical similarity by using different weighting methods.

Weighting Methods for Lexical Chains

- **LCNum:** number of the lexical chains which span the two posts.
- **LCLen:** sum of the number of tokens from lexical chains which span the two posts.
- **LCStr:** sum of the strength scores of lexical chains which span the two posts:

$$Strength(Chain) = Length \times Homogeneity$$

- **LCBan:** sum of the balance scores of lexical chains which span the two posts:

$$Balance(Chain) = \begin{cases} n_1/n_2 & if \ \ n_1 < n_2 \\ n_2/n_1 & else \end{cases}$$

Evaluation Metrics

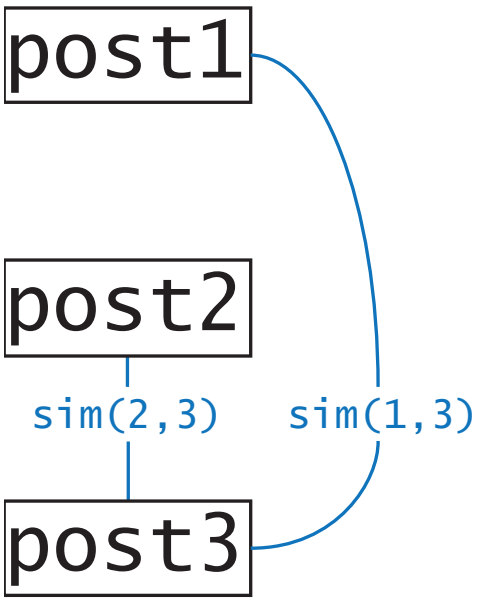
- Use micro-averaged Precision ($\mathcal{P}_\mu$), Recall ($\mathcal{R}_\mu$) and F-score ($\mathcal{F}_\mu$: $\beta = 1$)
- $\mathcal{F}_\mu$ is the main evaluation metric.
- An informed heuristic baseline (*Heuristic*) is used, where all first posts are labelled with link 0 and all other posts are labelled with link 1.
- Significance test: randomised estimation with $p < 0.05$

Reference: Yeh, 2000

Assumptions, Experiments and Analysis

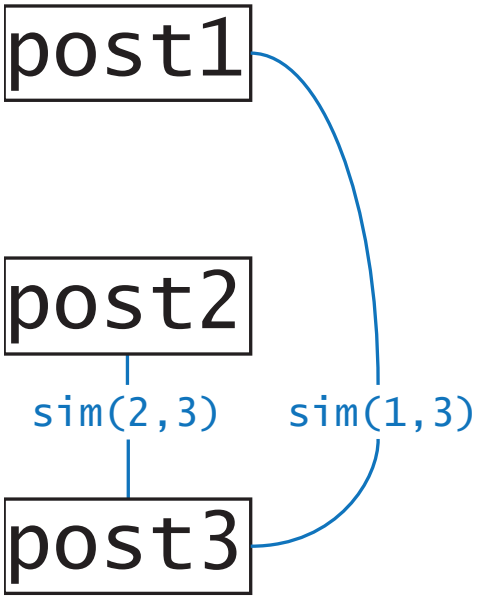
Initial Assumption and Experiments

- **Assumption 1:** A post should be similar to the preceding post it is linked to.



Initial Assumption and Experiments

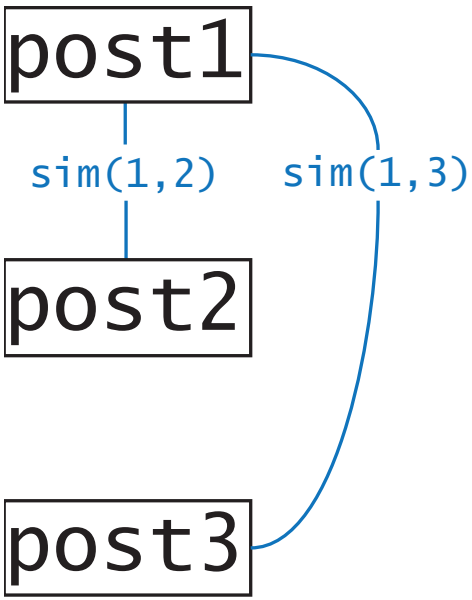
- **Assumption 1:** A post should be similar to the preceding post it is linked to.



Classifier	Weighting	$\mathcal{P}_\mu$	$\mathcal{R}_\mu$	$\mathcal{F}_\mu$
Heuristic	—	**.810**	**.772**	**.791**
$Chainer_{Roget}$	LCNum	.755	.720	.737
	LCLen	.737	.703	.720
	LCStr	.802	.764	.783
	LCBan	.723	.689	.706
$Chainer_{WN}$	LCNum	.685	.644	.660
	LCLen	.676	.651	.667
	LCStr	.718	.685	.701
	LCBan	.683	.651	.667
$Chainer_{SV}$	LCNum	.648	.618	.632
	LCLen	.630	.601	.615
	LCStr	.627	.598	.612
	LCBan	.645	.615	.630

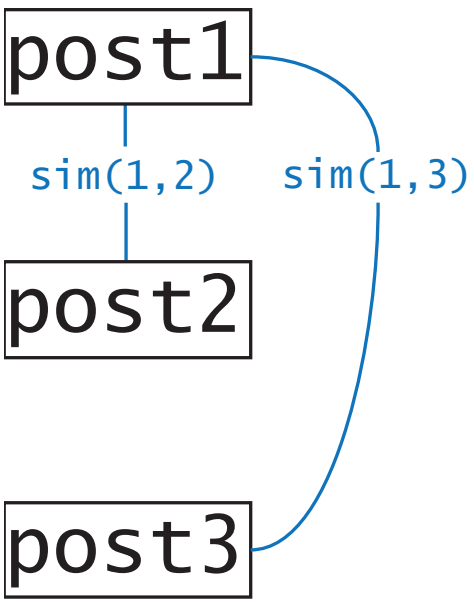
Post 3 Analysis

- **Assumption 2:** If the Post 3 vs. Post 1 lexical similarity is larger than Post 2 vs. Post 1 lexical similarity, then Post 3 is more likely to be linked back to Post 1.



Post 3 Analysis

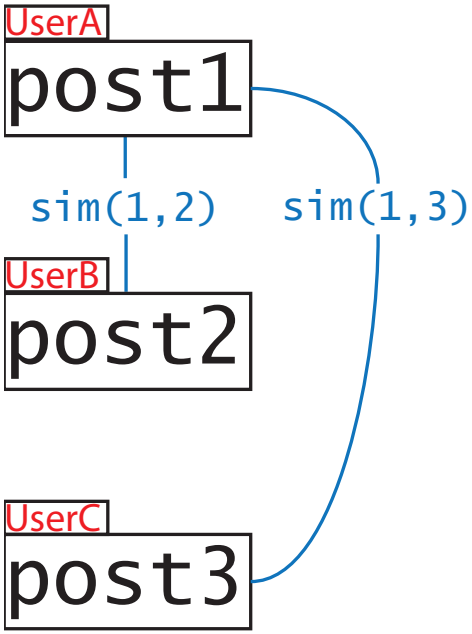
- **Assumption 2:** If the Post 3 vs. Post 1 lexical similarity is larger than Post 2 vs. Post 1 lexical similarity, then Post 3 is more likely to be linked back to Post 1.



Classifier	Weighting	$\mathcal{P}_\mu$	$\mathcal{R}_\mu$	$\mathcal{F}_\mu$
Heuristic	—	.810	.772	.791
Chainer$_{Roget}$	LCNum	.811	.773	.791
	LCLen	.811	.773	.791
	LCStr	.810	.772	.791
	LCBan	.813	.775	.794
Chainer$_{WN}$	LCNum	.806	.768	.786
	LCLen	.806	.769	.787
	LCStr	.806	.769	.787
	LCBan	.809	.771	.789
Chainer$_{SV}$	LCNum	.813	.775	.794
	LCLen	.813	.775	.794
	LCStr	.816	.778	.797
	LCBan	**.818**	**.780**	**.799**

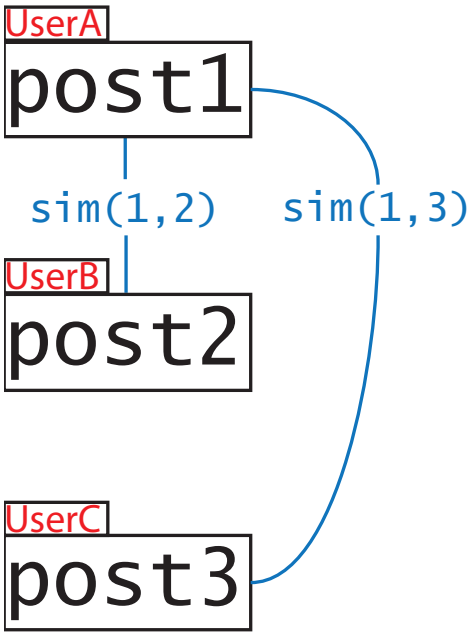
Post 3 Analysis . . .

- **Assumption 3:** If Post 3 vs. Post 1 lexical similarity is larger than Post 2 vs. Post 1 lexical similarity and Post 3 is not posted by the initiator of the thread, then Post 3 is more likely to be linked back to Post 1.



Post 3 Analysis . . .

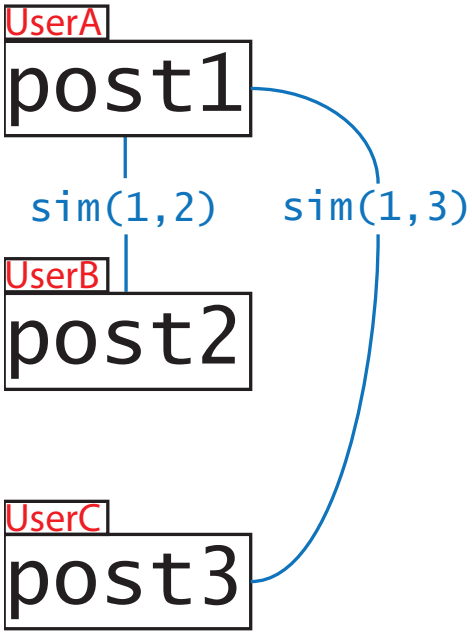
- **Assumption 3:** If Post 3 vs. Post 1 lexical similarity is larger than Post 2 vs. Post 1 lexical similarity and Post 3 is not posted by the initiator of the thread, then Post 3 is more likely to be linked back to Post 1.



Classifier	Weighting	$\mathcal{P}_\mu$	$\mathcal{R}_\mu$	$\mathcal{F}_\mu$
Heuristic	—	.810	.772	.791
Chainer$_{SV}$	LCNum	.832	.793	.812
	LCLen	.832	.793	.812
	LCStr	.831	.793	.812
	LCBan	**.836**	**.797**	**.816**

Post 3 Analysis . . .

- **Assumption 3:** If Post 3 vs. Post 1 lexical similarity is larger than Post 2 vs. Post 1 lexical similarity and Post 3 is not posted by the initiator of the thread, then Post 3 is more likely to be linked back to Post 1.

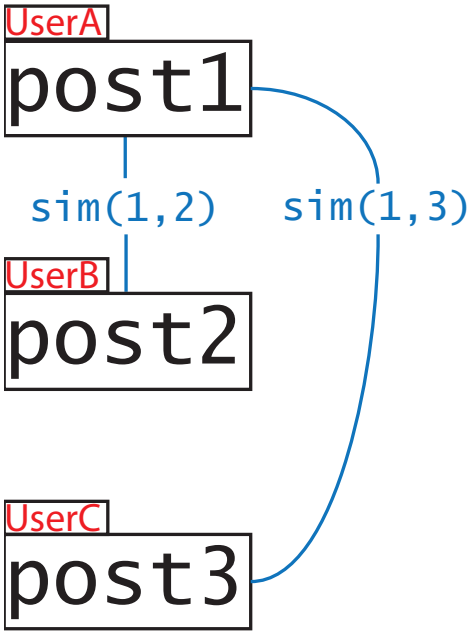


Classifier	Weighting	$\mathcal{P}_\mu$	$\mathcal{R}_\mu$	$\mathcal{F}_\mu$
Heuristic	—	.810	.772	.791
$Chainer_{SV}$	LCNum	.832	.793	.812
	LCLen	.832	.793	.812
	LCStr	.831	.793	.812
	LCBan	**.836**	**.797**	**.816**

★ $Heuristic_{user}$: Post 3 is linked to Post 1 if these two posts are from different users and all the other posts are linked as *Heuristic*.

Post 3 Analysis . . .

- **Assumption 3:** If Post 3 vs. Post 1 lexical similarity is larger than Post 2 vs. Post 1 lexical similarity and Post 3 is not posted by the initiator of the thread, then Post 3 is more likely to be linked back to Post 1.



Classifier	Weighting	$\mathcal{P}_\mu$	$\mathcal{R}_\mu$	$\mathcal{F}_\mu$
Heuristic	—	.810	.772	.791
$Heuristic_{user}$	—	**.839**	**.800**	**.819**
$Chainer_{SV}$	LCNum	.832	.793	.812
	LCLen	.832	.793	.812
	LCStr	.831	.793	.812
	LCBan	**.836**	**.797**	**.816**

★ $Heuristic_{user}$: Post 3 is linked to Post 1 if these two posts are from different users and all the other posts are linked as *Heuristic*.

Lexical Chaining for Supervised Learning

- Add a lexical chaining based feature to the classifier of Wang et al. [2011] based on Assumption 3:

$$feature = \begin{cases} \frac{sim(post3, post1)}{sim(post2, post1)} & Post3 \\ 0 & NonPost3 \end{cases}$$

Lexical Chaining for Supervised Learning

- Add a lexical chaining based feature to the classifier of Wang et al. [2011] based on Assumption 3:

$$feature = \begin{cases} \frac{sim(post3, post1)}{sim(post2, post1)} & Post3 \\ 0 & NonPost3 \end{cases}$$

- Experimental results:

Feature	Weighting	$\mathcal{P}_\mu$	$\mathcal{R}_\mu$	$\mathcal{F}_\mu$
Heuristic	—	.810	.772	.791
$Heuristic_{user}$	—	.839	.800	.819
NoLC	—	.898	.883	.891
WithLC	LCNum	.901	.886	.894
	LCLen	.902	.887	.894
	LCStr	.899	.884	.891
	LCBan	**.905**	**.890**	**.897**

Summary

- Explore unsupervised approaches for thread linking structure recovery, by automatically analysing the lexical cohesion between posts.
- Preliminary experiments derive results which are better than an informed baseline (i.e. *Heuristic*).
- The feature extracted from lexical chains also help the supervised method.
- Future work:
  - explore methods which can be used to recover all the inter-post links.
  - out-of-domain supervised learning vs. unsupervised learning
  - investigate more effective ways of extracting features from the created lexical chains to improve supervised learning.

Thank You!

Questions?

References I

Michel Galley and Kathleen McKeown. Improving word sense disambiguation in lexical chaining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 1486–1488, Acapulco, Mexico, 2003.

Mario Jarmasz and Stan Szpakowicz. The design and implementation of an electronic lexical knowledge base. *Advances in Artificial Intelligence*, 2056(2001):325–334, 2001.

Mario Jarmasz and Stan Szpakowicz. Not as easy as it seems: Automating the construction of lexical chains using roget's thesaurus. *Advances in Artificial Intelligence*, 2671(2003):994–999, 2003.

Su Nam Kim, Li Wang, and Timothy Baldwin. Tagging and linking web forum posts. In *Proceedings of the 14th Conference on Computational Natural Language Learning (CoNLL-2010)*, pages 192–202, Uppsala, Sweden, 2010.

Meghana Marathe and Graeme Hirst. Lexical chains using distributional measures of concept distance. In *Proceedings, 11th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2010)*, pages 291–302, Iași, Romania, 2010.

Hinrich Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1): 97–123, 1998.

Li Wang, Marco Lui, Su Nam Kim, Joakim Nivre, and Timothy Baldwin. Predicting thread discourse structure over technical web forums. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 13–25, Edinburgh, UK, 2011. URL `http://www.aclweb.org/anthology/D11-1002`.

References II

Dominic Widdows and Kathleen Ferraro. Semantic Vectors: a scalable open source package and online technology management application. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, pages 1183–1190, Marrakech, Morocco, 2008.

Alexander Yeh. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, pages 947–953, Saarbrücken, Germany, 2000.