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
li.wang.d@gmail.com, diana@dianamccarthy.co.uk, tb@ldwin.net

Abstract

Web user forums are valuable means for users to resolve specific information needs, both interactively for participants and statically for users who search/browse over historical thread data. However, the complex structure of forum threads can make it difficult for users to extract relevant information. Thread linking structure has the potential to help tasks such as information retrieval (IR) and threading visualisation of forums, thereby improving information access. Unfortunately, thread linking structure is not always available in forums.

This paper proposes an unsupervised approach to predict forum thread linking structure using lexical chaining, a technique which identifies lists of related word tokens within a given discourse. Three lexical chaining algorithms, including one that only uses statistical associations between words, are experimented with. Preliminary experiments lead to results which surpass an informed baseline.

1 Introduction

Web user forums (or simply "forums") are online platforms for people to discuss and obtain information via a text-based threaded discourse, generally in a pre-determined domain (e.g. IT support or DSLR cameras). With the advent of Web 2.0, there has been rapid growth of web authorship in this area, and forums are now widely used in various areas such as customer support, community development, interactive reporting and online education. In addition to providing the means to interactively participate in discussions or obtain/provide answers to questions, the vast volumes of data contained in forums make them a valuable resource for "support sharing", i.e. looking over records of past user interactions to potentially find an immediately applicable solution to a current problem. On the one hand, more and more answers to questions over a wide range of domains are becoming available on forums; on the other hand, it is becoming harder and harder to extract and access relevant information due to the sheer scale and diversity of the data.

Previous research shows that the thread linking structure can be used to improve information retrieval (IR) in forums, at both the post level (Xi et al., 2004; Seo et al., 2009) and thread level (Seo et al., 2009; Elsas and Carbonell, 2009). These inter-post links also have the potential to enhance threading visualisation, thereby improving information access over complex threads. Unfortunately, linking information is not supported in many forums. While researchers have started to investigate the task of thread linking structure recovery (Kim et al., 2010; Wang et al., 2011b), most research efforts focus on supervised methods.

To illustrate the task of thread linking recovery, we use an example thread, made up of 5 posts from 4 distinct participants, from the CNET forum dataset of Kim et al. (2010), as shown in Figure 1. The linking structure of the thread is modelled as a rooted directed acyclic graph (DAG). In this example, UserA initiates the thread with a question in the first post, by asking how to create an interactive input box on a webpage. This post is linked to a virtual root with link label 0. In response, UserB and UserC pro-

Li Wang, Diana Mccarthy and Timothy Baldwin. 2011. Predicting Thread Linking Structure by Lexical Chaining. In *Proceedings of Australasian Language Technology Association Workshop*, pages 76−85

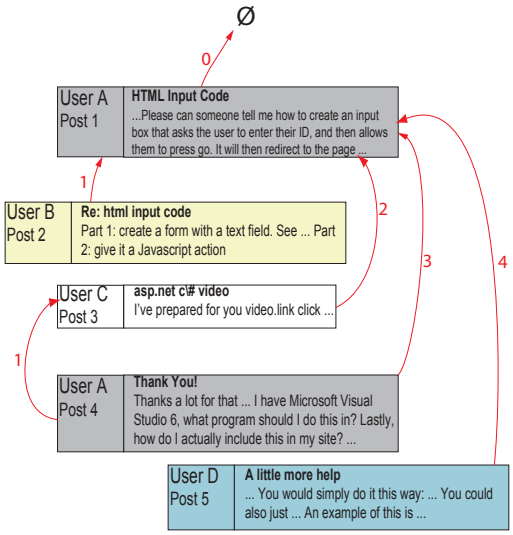

Figure 1: A snippeted CNET thread annotated with linking structure

vide independent answers. Therefore their posts are linked to the first post, with link labels 1 and 2 respectively. UserA responds to UserC (link = 1) to confirm the details of the solution, and at the same time, adds extra information to his/her original question (link = 3); i.e., this one post has two distinct links associated with it. Finally, UserD proposes a different solution again to the original question (link = 4).

Lexical chaining is a technique for identifying lists of related words (lexical chains) within a given discourse. The extracted lexical chains represent the discourse's lexical cohesion, or "cohesion indicated by relations between words in the two units, such as use of an identical word, a synonym, or a hypernym" (Jurafsky and Martin, 2008, pp. 685).

Lexical chaining has been investigated in many research tasks such as text segmentation (Stokes et al., 2004), word sense disambiguation (Galley and McKeown, 2003), and text summarisation (Barzilay and Elhadad, 1997). The lexical chaining algorithms used usually rely on domain-independent thesauri such as Roget's Thesaurus, the Macquarie Thesaurus (Bernard, 1986) and WordNet (Fellbaum, 1998), with some algorithms also utilising statistical associations between words (Stokes et al., 2004; Marathe and Hirst, 2010).

This paper explores unsupervised approaches for forum thread linking structure recovery, by using lexical chaining to analyse the inter-post lexical cohesion. We investigate three lexical chaining algorithms, including one that only uses statistical associations between words. The contributions of this research are:

- Proposal of an unsupervised approach using lexical chaining to recover the inter-post links in web user forum threads.
- Proposal of a lexical chaining approach that only uses statistical associations between words, which can be calculated from the raw text of the targeted domain.

The remainder of this paper is organised as follows. Firstly, we review related research on forum thread linking structure classification and lexical chaining. Then, the three lexical chaining algorithms used in this paper are described in detail. Next, the dataset and the experimental methodology are explained, followed by the experiments and analysis. Finally, the paper concludes with a brief summary and possible future work.

2 Related Work

The linking structure of web user forum threads can be used in tasks such as IR (Xi et al., 2004; Seo et al., 2009; Elsas and Carbonell, 2009) and threading visualisation. However, many user forums don't support the user input of linking information. Automatically recovering the linking structure of forum threads is therefore an interesting task, and has started to attract research efforts in recent years. All the methods investigated so far are supervised, such as ranking SVMs (Seo et al., 2009), SVM-HMMs (Kim et al., 2010), Maximum Entropy (Kim et al., 2010) and Conditional Random Fields (CRF) (Kim et al., 2010; Wang et al., 2011b; Wang et al., 2011a; Aumayr et al., 2011), with CRF models frequently being reported to deliver superior performance. While there is research that attempts to conduct cross-forum classification (Wang et al., 2011a) — where classifiers are trained over linking labels from one forum and tested over threads from other forums — the results have not been promising. This research explores unsupervised methods for thread

linking structure recovery, by exploiting lexical cohesion between posts via lexical chaining.

The first computational model for lexical chain extraction was proposed by Morris and Hirst (1991), based on the use of the hierarchical structure of Roget's International Thesaurus, 4th Edition (1977). Because of the lack of a machine-readable copy of the thesaurus at the time, the lexical chains were built by hand. Research in lexical chaining has then been investigated by researchers from different research fields such as information retrieval, and natural language processing. It has been demonstrated that the textual knowledge provided by lexical chains can benefit many tasks, including text segmentation (Kozima, 1993; Stokes et al., 2004), word sense disambiguation (Galley and McKeown, 2003), text summarisation (Barzilay and Elhadad, 1997), topic detection and tracking (Stokes and Carthy, 2001), information retrieval (Stairmand, 1997), malapropism detection (Hirst and St-Onge, 1998), and question answering (Moldovan and Novischi, 2002).

Many types of lexical chaining algorithms rely on examining lexicographical relationships (i.e. semantic measures) between words using domain-independent thesauri such as the Longmans Dictionary of Contemporay English (Kozima, 1993), Roget's Thesaurus (Jarmasz and Szpakowicz, 2003), Macquarie Thesaurus (Marathe and Hirst, 2010) or WordNet (Barzilay and Elhadad, 1997; Hirst and St-Onge, 1998; Moldovan and Novischi, 2002; Galley and McKeown, 2003). These lexical chaining algorithms are limited by the linguistic resources they depend upon, and often only apply to nouns.

Some lexical chaining algorithms also make use of statistical associations (i.e. distributional measures) between words which can be automatically generated from domain-specific corpora. For example, Stokes et al. (2004)'s lexical chainer extracts significant noun bigrams based on the G^2 statistic (Pedersen, 1996), and uses these statistical word associations to find related words in the preceding context, building on the work of Hirst and St-Onge (1998). Marathe and Hirst (2010) use distributional measures of conceptual distance, based on the methodology of Mohammad and Hirst (2006) to compute the relation between two words. This framework uses a very coarse-grained sense (concept or category) inventory from the Macquarie Thesaurus (Bernard, 1986) to build a word-category co-occurrence matrix (WCCM), based on the British National Corpus (BNC). Lin (1998a)'s measure of distributional similarity based on point-wise mutual information (PMI) is then used to measure the association between words.

This research will explore two thesaurus-based lexical chaining algorithms, as well as a novel lexical chaining approach which relies solely on statistical word associations.

3 Lexical Chaining Algorithms

Three lexical chaining algorithms are experimented with in this research, as detailed in the following sections.

3.1 $Chainer_{Roget}$

$Chainer_{Roget}$ is a Roget's Thesaurus based lexical chaining algorithm (Jarmasz and Szpakowicz, 2003) based on an off-the-shelf package, namely the Electronic Lexical Knowledge Base (ELKB) (Jarmasz and Szpakowicz, 2001).

The underlying methodology of $Chainer_{Roget}$ is shown in Algorithm 1. Methods used to calculate the chain strength/weight are presented in Section 5. While the original Roget's Thesaurus-based algorithm by Morris and Hirst (1991) proposes five types of thesaural relations to add a candidate word in a chain, $Chainer_{Roget}$ only uses the first one, as is explained in Algorithm 1. Moreover, while Jarmasz and Szpakowicz (2003) use the 1987 Penguin's Roget's Thesaurus in their research, the ELKB package uses the Roget's Thesaurus from 1911 due to copyright restriction.

3.2 $Chainer_{WN}$

$Chainer_{WN}$ is a non-greedy WordNet-based chaining algorithm proposed by Galley and McKeown (2003). We reimplemented their method based on an incomplete implementation in NLTK.[1]

The algorithm of $Chainer_{WN}$ is based on the assumption of one sense per discourse, and can be decomposed into three steps. Firstly, a "disambiguation graph" is built by adding the candidate nouns of

[1] http://people.virginia.edu/~ma5ke/classes/files/cs65lexicalChain.pdf

Algorithm 1 $Chainer_{Roget}$

select a set of candidate nouns
for each candidate noun **do**
build all the possible chains, where each pair of nouns in each chain are either the same word or included in the same *Head* of *Roget's Thesaurus*, and select the strongest chain for each candidate noun.
end for
merge two chains if they contain at least one noun in common

the discourse one by one. Each node in the graph represents a noun instance with all its senses, and each weighted edge represents the semantic relation between two senses of two nouns. The weight of each edge is calculated based on the distances between nouns in the discourse. Secondly, word sense disambiguation (WSD) is performed. In this step, a score of every sense of each noun node is calculated by summing the weight of all edges leaving that sense. The sense of each noun node with the highest score is considered as the right sense of this noun in the discourse. Lastly, all the edges of the disambiguation graph connecting (assumed) wrong senses of every noun node are removed, and the remaining edges linking noun nodes form the lexical chains of the discourse. The semantic relations exploited in this algorithm include hypernyms/hyponyms and siblings (i.e. hyponyms of hypernyms).

3.3 $Chainer_{SV}$

$Chainer_{SV}$, as shown in Algorithm 2, is adapted from Marathe and Hirst (2010)'s lexical chaining algorithm. The main difference between $Chainer_{SV}$ and the original algorithm is the method used to calculate associations between words. Marathe and Hirst (2010) use two different measures, including Lin (1998b)'s WordNet-based measure, and Mohammad and Hirst (2006)'s distributional measures of concept distance framework. In $Chainer_{SV}$, we use word vectors from WORDSPACE (Schütze, 1998) models and apply cosine similarity to compute the associations between words. WORDSPACE is a multi-dimensional real-valued space, where words, contexts and senses are represented as vectors. A vector for word w is derived from words that co-occur with w. A dimensionality reduction technique is often used to reduce the dimension of the vector. We build the WORDSPACE model with SemanticVectors (Widdows and Ferraro, 2008), which is based on Random Projection dimensionality reduction (Bingham and Mannila, 2001).

The underlying methodology of $Chainer_{SV}$ is shown in Algorithm 2. This algorithm requires a method to calculate the similarity between two tokens (i.e. words): $sim_{tt}(x, y)$, which is done by computing the cosine similarity of the two tokens' semantic vectors. The similarity between a token t_i and a lexical chain c_j is then calculated by:

$$sim_{tc}(t_i, c_j) = \sum_{t_k \in c_j} \frac{1}{l_j} sim_{tt}(t_i, t_k)$$

where l_j represents the length of lexical chain c_j. The similarity between two chains c_i and c_j is then computed by:

$$sim_{cc}(c_i, c_j) = \sum_{t_m \in c_i, t_n \in c_j} \frac{1}{l_i \times l_j} sim_{tt}(t_m, t_n)$$

where l_i and l_j are the lengths of c_i and c_j respectively.

As is shown in Algorithm 2, $Chainer_{SV}$ has two parameters: the threshold for adding a token to a chain, $threshold_a$; and the threshold for merging two chains, $threshold_m$. A larger $threshold_a$ leads to conservative chains where tokens in a chain are strongly related, while a smaller $threshold_a$ results in longer chains where the relationship between tokens in a chain may not be clear. Similarly, a larger $threshold_m$ is conservative and leads to less chain merging, while a smaller $threshold_m$ may create longer but less meaningful chains. Our initial experiments show that the combination of $threshold_a = 0.1$ and $threshold_m = 0.05$ often results in lexical chains with reasonable lengths and interpretations. Therefore, this parameter setting will be used throughout all the experiments described in this paper.

4 Task Description and Dataset

The main task performed in this research is to recover inter-post links within forum threads, by

Algorithm 2 $Chainer_{SV}$

$chains = empty$
select a set of candidate tokens
for each candidate token t_i **do**
  $max\_score = \max_{c_j \in chains}(sim_{tc}(t_i, c_j))$
  $max\_chain = \arg\max_{c_j \in chains}(sim_{tc}(t_i, c_j))$
  if $chains = empty$ **or** $max\_score < threshold_a$ **then**
    create a new chain c_k containing t_i and add c_k to $chains$
  else if more than one max_chain **then**
    merge chains if the two chains' similarity is larger than $threshold_m$, and add t_i to the resultant chain or the first max_chain
  else
    add t_i to the max_chain
  end if
end for
return $chains$

analysing the lexical chains extracted from the posts. In this, we assume that a post can only link to an earlier post (or a virtual root node). Following Wang et al. (2011b), it is possible for there to be multiple links from a given post, e.g. if a post both confirms the validity of an answer and adds extra information to the original question (as happens in Post4 in Figure 1).

The dataset we use is the CNET forum dataset of Kim et al. (2010),[2] which contains 1332 annotated posts spanning 315 threads, collected from the Operating System, Software, Hardware and Web Development sub-forums of CNET.[3] Each post is labelled with one or more links (including the possibility of null-links, where the post doesn't link to any other post), and each link is labelled with a dialogue act. We only use the link part of the annotation in this research. For the details of the dialogue act tagset, see Kim et al. (2010).

We also obtain the original crawl of CNET forum collected by Kim et al. (2010), which contains 262,402 threads. To build a WORDSPACE model for $Chainer_{SV}$ as is explained in Section 3, only the threads from the four sub-forums mentioned above are chosen, which consist of 536,482 posts spanning 114,139 threads. The reason for choosing only a subset of the whole dataset is to maintain the same types of technical dialogues as the annotated posts. The texts (with stop words and punctuations removed) from the titles and bodies of the posts are then extracted and fed into the SemanticVectors package with default settings to obtain the semantic vector for each word token.

[2]Available from `http://www.csse.unimelb.edu.au/research/lt/resources/conll2010-thread/`

[3]`http://forums.cnet.com/`

5 Methodology

To the best of our knowledge, no previous research has adopted lexical chaining to predict inter-post links. The basic idea of our approach is to use lexical chains to measure the inter-post lexical cohesion (i.e. lexical similarity), and use these similarity scores to reconstruct inter-post links. To measure the lexical cohesion between two posts, the texts (with stop words and punctuations removed) from the titles and bodies of the two posts are first combined. Then, lexical chainers are applied over the combined texts to extract lexical chains. Lastly, the following weighting methods are used to calculate the lexical similarity between the two posts:

LCNum: the number of the lexical chains which span the two posts.

LCLen: find the lexical chains which span the two posts, and use the sum of tokens contained in each as the similarity score.

LCStr: find the lexical chains which span the two posts, and use the sum of each chain's chain strength as the similarity score. The chain strength is calculated by using a formula suggested by Barzilay and Elhadad (1997):

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

where $Length$ is the number of tokens in the chain, and $Homogeneity$ is $1-$ the number of distinct token occurrences divided by the $Length$.

LCBan: find the lexical chains which span the two posts, and use the sum of each chain's balance score as the similarity score. The balance score

is calculated by using the following formula:

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

where n_1 is the number of tokens from the chain belonging to the first post, and n_2 is the number of tokens from the chain belonging to the second post.

6 Assumptions, Experiments and Analysis

The experiment results are evaluated using micro-averaged Precision ($\mathcal{P}_\mu$), Recall ($\mathcal{R}_\mu$) and F-score ($\mathcal{F}_\mu$: $\beta = 1$), with $\mathcal{F}_\mu$ as the main evaluation metric. The statistical significance is tested using randomised estimation (Yeh, 2000) with $p < 0.05$.

As our baseline for the unsupervised task, an informed heuristic ($Heuristic$) is used, where all first posts are labelled with link 0 (i.e. link to a virtual root) and all other posts are labelled with link 1 (i.e. link to the immediately preceding post).

As is explained in Section 4, it is possible for there to be multiple links from a given post. Because these kinds of posts, which only account for less than 5% of the total posts, are sparse in the dataset, we only consider recovering one link per post in our experiments. However, our evaluation still considers all links (meaning that it is not possible for our methods to achieve an F-score of 1.0).

6.1 Initial Assumption and Experiments

We observe that in web user forum threads, if a post replies to a preceding post, the two posts are usually semantically related and lexically similar. Based on this observation, we make the following assumption:

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

This assumption leads to our first unsupervised model, which compares each post (except for the first and second) in a given thread with all its preceding posts one by one, by firstly identifying the lexical chains using the lexical chainers described in Section 3 and then calculating the inter-post lexical similarity using the methods explained in Section 5. The experimental results are shown in Table 1.

From Table 1 we can see that no results surpass the $Heuristic$ baseline. Further investigation reveals that while Assumption 1 is reasonable, it is not always correct —i.e. similar posts are not always linked together. For example, an answer post later in a thread might be linked back to the first question post but be more similar to preceding answer posts, to which it is not linked, simply because they are all answers to the same question. The initial experiments show that more careful analysis is needed to use inter-post lexical similarity to reconstruct inter-post linking.

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

Table 1: Results from the Assumption 1 based unsupervised approach, by using three lexical chaining algorithms with four different weighting schemes.

6.2 Post 3 Analysis

Because Post 1 and Post 2 are always labelled with link 0 and 1 respectively, our analysis starts from Post 3 of each thread. Based on the analysis, the second assumption is made:

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

Assumption 2 leads to an unsupervised approach which combines the three lexical chaining algorithms introduced in Section 3 with the four weighting schemes explained in Section 5 to measure Post 3 vs. Post 1 similarity and Post 2 vs. Post 1 similarity. If the former is larger, Post 3 is linked back to Post 1, otherwise Post 3 is linked back to Post 2. As for the other posts, the link labels are the same as the ones from the $Heuristic$ baseline. The experimental results are shown in Table 2.

From the results in Table 2 we can see that $Chainer_{SV}$ is the only lexical chaining algorithm

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

Table 2: Results from the Assumption 2 based unsupervised approach, by using three lexical chaining algorithms with four different weighting schemes.

that leads to results which are better than the $Heuristic$ baseline. Analysis over the lexical chains generated by the three lexical chainers shows that both $Chainer_{Roget}$ and $Chainer_{WN}$ extract very few chains, most of which contain only repetitions of a same word. This is probably because these two lexical chainers only consider nouns, and therefore have limited input tokens. Especially for $Chainer_{Roget}$ which uses an old dictionary (1911 edition) that does not contain modern technical terms, such as *Windows*, *OSX* and *PC*. While $Chainer_{WN}$ uses WordNet which has a larger and more modern vocabulary, the chainer considers very limited semantic relations (i.e. hypernyms, hyponyms and hyponyms of hypernyms). Moreover, the texts in forum posts are usually relatively short and informal, and contain typos and non-standard acronyms. These factors make it very difficult for $Chainer_{Roget}$ and $Chainer_{WN}$ to extract lexical chains. As for $Chainer_{SV}$, because all the words (except for stop words) are considered as candidate words, and relations between words are flexible according to the thresholds (i.e. $threshold_a$ and $threshold_m$), relatively abundant lexical chains are generated. While some of the chains clearly capture lexical cohesion among words, some of the chains are hard to interpret. Nevertheless, the results from $Chainer_{SV}$ are encouraging for the unsupervised approach, and therefore further investigation is conducted using only $Chainer_{SV}$.

Because the experiments based on the Assumption 2 derive promising results, further analysis is conducted to enforce this assumption. We notice that the posts from the initiator of a thread are often outliers compared to other posts — i.e. these posts are similar to the first post because they are from the same author, but at the same time an initiator rarely replies to his/her own posts. This observation leads to a stricter assumption:

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

Table 3: Results from the Assumption 3 based unsupervised approach, by using $Chainer_{SV}$ with different weighting schemes

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.*

Based on Assumption 3, experiments are carried out using $Chainer_{SV}$ with different weighting schemes. We also introduce a stronger baseline ($Heuristic_{user}$) based on Assumption 3, where Post 3 is linked to Post 1 if these two posts are from different users and all the other posts are linked as $Heuristic$. The experimental results are shown in Table 3.

From Table 3 we can see that while all the results from $Chainer_{SV}$ are significantly better than the result from the $Heuristic$ baseline, with the LCBan weighting leading to the best $\mathcal{F}_\mu$ of 0.816, these results are not significantly different from the $Heuristic_{user}$ baseline. It is clear that the improvements attribute to the user constraint introduced in Assumption 3. This observation matches up with the results of supervised classification from Wang et al. (2011b), where the benefits brought by text similarity based features (i.e. TitSim and PostSim) are covered by more effective user information based features (i.e. UserProf).

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

Table 4: Supervised linking classification by applying $CRFSGD$ over features from Wang et al. (2011b) without (NoLC) and with (WithLC) features extracted from lexical chains, created by $Chainer_{SV}$ with different weighting schemes

6.3 Lexical Chaining for Supervised Learning

It is interesting to see whether our unsupervised approach can contribute to the supervised methods by providing additional features. To test this idea, we add a lexical chaining based feature to the classifier of Wang et al. (2011b) based on Assumption 3. The feature value for each post is calculated using the following formula:

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

where sim is calculated using $Chainer_{SV}$ with different weighting methods.

The experimental results are shown in Table 4. From the results we can see that, by adding the additional feature extracted from lexical chains, the results improve slightly. The feature from the $Chainer_{SV}$ with LCBan weighting leads to the best $\mathcal{F}_\mu$ of 0.897. These improvements are statistically insignificant, possibly because the information introduced by the lexical chaining feature is already captured by existing features. It is also possible that better feature representations are needed for the lexical chains.

These results are preliminary but nonetheless suggest the potential of utilising lexical chaining in the domain of web user forums.

6.4 Experiments over All the Posts

To date, all experiments have been based on just the first three posts in a thread, where the majority of our threads contain more than just three posts. We carried out preliminary experiments over full thread data, by generalising Assumption 3 to Post N for $N \geq 3$. However, no significant improvements were achieved over an informed baseline with our unsupervised approach. This is probably because the situation for later posts (after Post 3) is more complicated, as more linking options are possible. Relaxing the assumptions entirely also led to disappointing results. What appears to be needed is a more sophisticated set of constraints, to generalise the assumptions made for Post 3 to all the posts. We leave this for future work.

7 Conclusion

Web user forums are a valuable information source for users to resolve specific information needs. However, the complex structure of forum threads poses a challenge for users trying to extract relevant information. While the linking structure of forum threads has the potential to improve information access, these inter-post links are not always available.

In this research, we explore unsupervised approaches for thread linking structure recovery, by automatically analysing the lexical cohesion between posts. Lexical cohesion between posts is measured using lexical chaining, a technique to extract lists of related word tokens from a given discourse. Most lexical chaining algorithms use domain-independent thesauri and only consider nouns. In the domain of web user forums, where the texts of posts can be very short and contain various typos and special terms, these conventional lexical chaining algorithms often struggle to find proper lexical chains. To address this problem, we proposed the use of statistical associations between words, which are captured by the WORDSPACE model, to construct lexical chains. Our preliminary experiments derive results which are better than an informed baseline.

In future work, we want to explore methods which can be used to recover all the inter-post links. First, we plan to conduct more detailed analysis over inter-post lexical cohesion, and its relationship with inter-post links. Second, we want to investigate human linking behaviour in web user forums, hoping to find significant linking patterns. Furthermore, we want to investigate more methods and resources for constructing lexical chains, e.g. Cramer et al. (2012).

On top of exploring these potential approaches, it is worth considering stronger baseline methods such as using cosine similarity to measure inter-post similarity.

The $Chainer_{SV}$, as described in Section 4, is built on a WORDSPACE model learnt over a subset of four domains. It is also worth comparing with a more general WORDSPACE model learnt over the whole dataset.

As for supervised learning, it would be interesting to conduct experiments out of domain (i.e. train the model over threads from one forum, and classify threads from another forum), and compare with the unsupervised approaches. We also hope to investigate more effective ways of extracting features from the created lexical chains to improve supervised learning.

Acknowledgements

The authors wish to thank Malcolm Augat and Margaret Ladlow for providing access to their lexical chaining code, which was used to implement $Chainer_{WN}$. NICTA is funded by the Australian government as represented by Department of Broadband, Communication and Digital Economy, and the Australian Research Council through the ICT Centre of Excellence programme.

References

Erik Aumayr, Jeffrey Chan, and Conor Haye. 2011. Reconstruction of threaded conversations in online discussion forums. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM-11)*, pages 26–33, Barcelona, Spain.

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *Proceedings of the Intelligent Scalable Text Summarization Workshop*, pages 10–17, Madrid, Spain.

J.R.L. Bernard, editor. 1986. *The Macquarie Thesaurus*. Macquarie Library,, Sydney, Australia.

Ella Bingham and Heikki Mannila. 2001. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '01)*, pages 245–250, San Francisco, USA.

Léon Bottou. 2011. CRFSGD software. `http://leon.bottou.org/projects/sgd`.

Irene Cramer, Tonio Wandmacher, and Ulli Waltinger. 2012. Exploring resources for lexical chaining: A comparison of automated semantic relatedness measures and human judgments. In Alexander Mehler, Kai-Uwe Kühnberger, Henning Lobin, Harald Lüngen, Angelika Storrer, and Andreas Witt, editors, *Modeling, Learning, and Processing of Text Technological Data Structures*, volume 370 of *Studies in Computational Intelligence*, pages 377–396. Springer Berlin, Heidelberg.

Jonathan L. Elsas and Jaime G. Carbonell. 2009. It pays to be picky: An evaluation of thread retrieval in online forums. In *Proceedings of 32nd International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09)*, pages 714–715, Boston, USA.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge, USA.

Michel Galley and Kathleen McKeown. 2003. 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.

Graeme Hirst and David St-Onge. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum, editor, *WordNet: An electronic lexical database*, pages 305–332. The MIT Press, Cambridge, USA.

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

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

Daniel Jurafsky and James H. Martin. 2008. *SPEECH and LANGUAGE PROCESSING: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Prentice Hall, 2nd edition.

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

Hideki Kozima. 1993. Text segmentation based on similarity between words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 286–288, Columbus, USA.

Dekang Lin. 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th An-*

nual Meeting of the ACL and 17th International Conference on Computational Linguistics (COLING/ACL-98), pages 768–774, Montreal, Canada.

Dekang Lin. 1998b. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning (ICML'98)*, pages 296–304, Madison, USA.

Meghana Marathe and Graeme Hirst. 2010. 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, Iaşi, Romania.

Saif Mohammad and Graeme Hirst. 2006. Distributional measures of concept-distance: A task-oriented evaluation. In *Proceedings, 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 35–43, Sydney, Australia.

Dan Moldovan and Adrian Novischi. 2002. Lexical chains for question answering. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taiwan.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

Ted Pedersen. 1996. Fishing for exactness. In *Proceedings of the South-Central SAS Users Group Conference (SCSUG-96)*, Austin, USA.

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

Jangwon Seo, W. Bruce Croft, and David A. Smith. 2009. Online community search using thread structure. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*, pages 1907–1910, Hong Kong, China.

Mark A. Stairmand. 1997. Textual context analysis for information retrieval. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '97)*, pages 140–147, Philadelphia, USA.

Nicola Stokes and Joe Carthy. 2001. Combining semantic and syntactic document classifiers to improve first story detection. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2001)*, pages 424–425, New Orleans, USA.

Nicola Stokes, Joe Carthy, and Alan F. Smeaton. 2004. SeLeCT: a lexical cohesion based news story segmentation system. *AI Communications*, 17(1):3–12.

Hongning Wang, Chi Wang, ChengXiang Zhai, and Jiawei Han. 2011a. Learning online discussion structures by conditional random fields. In *Proceedings of the 34th Annual International ACM SIGIR Conference (SIGIR 2011)*, pages 435–444, Beijing, China.

Li Wang, Marco Lui, Su Nam Kim, Joakim Nivre, and Timothy Baldwin. 2011b. 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.

Dominic Widdows and Kathleen Ferraro. 2008. 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.

Wensi Xi, Jesper Lind, and Eric Brill. 2004. Learning effective ranking functions for newsgroup search. In *Proceedings of 27th International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)*, pages 394–401. Sheffield, UK.

Alexander Yeh. 2000. 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.