國立臺灣大學電機資訊學院網路與多媒體研究所

碩士論文

Graduate Institute of Networking and Multimedia
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

利用語彙、句法以及語義資訊偵測網路抄襲

Online Plagiarized Detection Through Exploiting Lexical, Syntactic, and Semantic Information

林琬瑜

Lin, Wan-Yu

指導教授：林守德 博士

Advisor: Shou-De Lin, Ph.D.

中華民國 102 年 1 月

Jan, 2013

Acknowledgements

這篇碩論、這份研究的完成要感謝許多人，逐段介紹如下。

第一個想感謝的，就是推薦這個題目給我的老師——林守德教授，不間斷的給予指導協助，在感覺辛苦、挫折的時候適時給予鼓勵，倘若少了老師的支持與鞭策，就不會有這篇論文，我也不會有機會因為這份研究，被計算語言學界最頂尖的會議 ACL-2012 邀請到韓國濟州島做系統展示。

顏君釗學長，謝謝你在一開始就為抄襲偵測系統的程式碼建立起良好的架構，為了將系統成功交接給我，陪我一路改程式改到入伍前一晚的凌晨三點多；之後只要我有不懂的地方，學長也總是盡速的給予支援和詳解；去年投稿 ACL-2011 short paper，也要多虧學長的鼎力襄助。雖然去年的投稿沒有被接受，今年的二度挑戰總算是成功達陣了，總之，一切的一切都非常感謝學長。

蔡青樺同學，很感謝妳當初邀我一同加入 mslab、一起投稿國科會大專生參與專題研究計劃。如果沒有妳的邀請，我現在還不知道會在那裡，應該也沒有機會接受守德老師的推薦來從事這份研究。

來自北京大學的訪問生——彭楠贇同學，謝謝親切、積極而知識淵博的妳，在我碩二這年給予的一切幫忙、支持和鼓勵，除了協助英文部分的實驗以及想法提供，今年投稿 ACL-2012 時論文撰寫，也都多虧有妳一起並肩作戰。沒有妳的參與，這篇論文能被 ACL 接受的機率肯定大大降低。

余守壹學長，謝謝你在我有點崩潰的碩論撰寫時期，給予免費無償還專業可靠的英文檢查，還有在口試講解流程、論文邏輯結構、系統架構圖設計等各大方面給予各項有用而具體的建議。倘若沒有你的諸多建議、細心檢查，我的碩士論文內容肯定無法像現在這樣流暢而有組織。

郭智中學弟，感謝你在接近 ACL demo & short paper 的投稿截止日，在我需要很多位址 IP、面臨實驗跑不完的窘境時，願意花時間幫忙整理資料、還借我很多台機器跑實驗。少了你的幫忙、以及當初某些崩潰時候給予的加油打氣，最終結果肯定難以想像！

楊政倫同學，真的要大大感謝你犧牲自己的研究和碩論撰寫時間，早起幫忙檢查、修改我們的 short 和 demo paper，提供專業的母語英文協助！

薛琇文同學、馮俊松同學、陳耀男同學、郭智中學弟，謝謝你們願意百忙中抽空來聽我的口試預演。從投影片的設計撰寫，到口頭講解的順序與邏輯修正，給了我許多可靠而精闢的建議，7 月 6 日下午的最終試煉——碩論口試能如此順利，你們功不可沒！

以及幫忙中文實驗標記的 63 位實驗者，如果沒有你們的幫忙，單靠我一個人是無法達成的，再次感謝你們願意臨危受命、傾囊相助。

還有已經有多年革命情感的實驗室夥伴們，meeting 時給予的檢討、生活上給予的陪伴，特別是李政德學長、解巽評學長、翁睿妤學姐等在碩論口室告急期間，多次給予的精神上和實際上的支持與鼓勵，除了感謝還是感謝。

最後還要感謝我的家人，尤其是我最愛的弟弟。因為有你們長久以來的支持與關懷，才有能順利取得台大碩士學位的今天的我。

摘要

傳統的抄襲偵測系統，許多只著重在文章的語彙統計特徵，至多再考慮句法結構，或利用 WordNet 來擷取文章的語義面訊息，且以離線的抄襲偵測居多；我們的系統則是將搜尋引擎整合進來，同時引進語彙、句法和語義這三個層面的結構特徵，抽取可疑文句組對裡，語彙的重覆率、重組率、連續性，單詞在句中所屬的詞性和片語標籤，以及透過 Latent Dirichlet Allocation (LDA) 所標記出的潛在主題來代表可能蘊含的語義資訊，如此結合這六個不同的抄襲偵測模型，再利用我們所設計的加權方法將六個模型的預測結果合併，是一個能自動偵測網路抄襲的線上系統。實驗結果顯示無論是英文還是中文的文章，我們的系統都能成功偵測出相當數量的可能抄襲來源，實驗數據上的表現也相較目前一些最先進的演算法還要來得突出。

關鍵字: 抄襲偵測, 語彙, 句法, 語義

Abstract

In this paper, we introduce a framework that identifies sentence and document level online plagiarism by exploiting lexical, syntactic and semantic features, which includes duplication ngram, reordering and alignment of words, POS and phrase tags, and semantic similarity of sentences. We also enhance plagiarism detection by establishing an ensemble framework to combine the prediction scores of each model. Experiments performed on English and Chinese corpora demonstrate that our system can not only find considerable amount of real-world online plagiarism cases but also outperforms several state-of-the-art algorithms.

Keywords: Plagiarism Detection, Lexical, Syntactic, Semantic

Contents

List of Figures

List of Tables

Chapter 1 Introduction

Online plagiarism, the action of trying to create a new piece of writing by copying, reorganizing or rewriting others' work identified through search engines, is one of the most commonly seen misusage of the highly matured Web technologies. As implied by the experiment conducted by ([2], Braumoeller and Gaines, 2001), a powerful plagiarism detection system can effectively discourage people from plagiarizing others' work. However, the definition of plagiarism is broad. In this work we try to focus on external plagiarism with translated plagiarism excluded. We further divide external plagiarism into two sub-types, verbatim plagiarism and smart plagiarism. The definitions are illustrated below in Table 1.

Table 1. Definition of verbatim plagiarism and smart plagiarism

Verbatim Plagiarism	Identical sentences are found between the two compared articles.
Smart Plagiarism	In the two compared articles, there exist certain sentence pairs which are not entirely identical, but are similar at the lexical level, or have similar syntactic structure or semantic meaning.

A common strategy people adopt for online-plagiarism detection is as follows. First they identify several suspicious sentences from the write-up and feed them one by one as a query to a search engine to obtain a set of documents. Then human reviewers can manually examine whether these documents are truly the sources of the suspicious sentences. While it is quite straightforward and effective, the limitation of this strategy is obvious. First, since the length of search query is limited, suspicious sentences are usually queried and examined independently. Therefore, it is harder to identify document level plagiarism than sentence level plagiarism. Second, manually checking whether a

query sentence plagiarizes certain websites requires specific domain and language knowledge as well as considerable amount of energy and time. To overcome the above shortcomings, we introduce an online plagiarism detection system using natural language processing (NLP) techniques to simulate the above reverse-engineering approach. We develop an ensemble framework that integrates lexical, syntactic and semantic features to achieve this goal. Our system is nearly language independent and we have implemented both English and Chinese versions for evaluation. Evaluation on English and Chinese datasets show that our system is effective and can consistently outperform state-of-the-art methods.

Chapter 2 Related Work

Plagiarism detection has been widely discussed in the past decades ([16], Zou et al., 2010). Table 2. summarizes some of them:

Table 2. Summary of related works

Author	Comparison Unit	Lexical Feature	Syntactic Feature	Semantic Feature	Similarity Function
Brin et al., 1995, [3]	Sentence	∨	x	x	Percentage of matching sentences.
White and Joy, 2004, [15]	Sentence	∨	x	x	Average overlap ratio of the sentence pairs using 2 pre-defined thresholds.
Niezgoda and Way, 2006, [9]	A human defined sliding window	∨	x	x	Sliding windows ranked by the average length per word.
Cedeno and Rosso, 2009, [4]	Sentence	∨	x	x	Overlap percentage of ngrams in the sentence pairs.
Pera and Ng, 2010, [10]	Sentence	∨	x	∨	Calculate average all pair word similarity as the overall sentence similarity using WordNet and word co-occurrence in Wekipedia.
Grman and Ravas, 2011, [5]	Passage	∨	x	∨	Overlap percentage of words with given thresholds on both ratio and absolute number of words in passage.
Stamatatos, 2011, [13]	Passage	∨	∨	x	Overlap percentage of stopword ngrams.
Ours, 2012, [8]	Sentence + Passage	∨	∨	∨	Accumulate ensemble scores of 6 lexical, syntactic, or semantic models with a pre-defined threshold for document level plagiarism detection.

From Table 2, we can see that the comparison units for many systems are limited to just sentences. For the systems which focus on passages, they lack the usage of either syntactic or semantic information, and only try to judge the similarity between the compared passages mainly by the overlap percentage of words or ngrams. In contrast, our

system deals with both sentence and passage level data and exploit lexical, syntactic and semantic information through the six proposed models to simulate what plagiarists are trying to do, and thus making our system more robust and reliable.

There are several online or charged/free downloadable plagiarism detection systems such as Turnitin[1], EVE2[2], Docol©c[3], and CATPPDS[4] which detect mainly verbatim copy. Others such as Microsoft Plagiarism Detector[5] (MPD), SafeAssign[6], Copyscape[7] and VeriGuide[8], claim to be capable of detecting obfuscations. Unfortunately those commercial systems do not reveal the detail strategies used, therefore it is hard to judge and reproduce their results for comparison.

[1] Turnitin: http://turnitin.com/

[2] EVE2: http://www.canexus.com/

[3] Docol© c: http://www.docoloc.de/

[4] CATPPDS: http://checker.cm.nsysu.edu.tw

[5] MPD: http://plagiarism-detector.com/

[6] SafeAssign: http://www.itap.purdue.edu/tlt/safeassign/index.cfm

[7] Copyscape: http://www.copyscape.com/

[8] VeriGuide: http://veriguide1.cse.cuhk.edu.hk/portal/plagiarism_detection/index.jsp

Chapter 3 Methodology

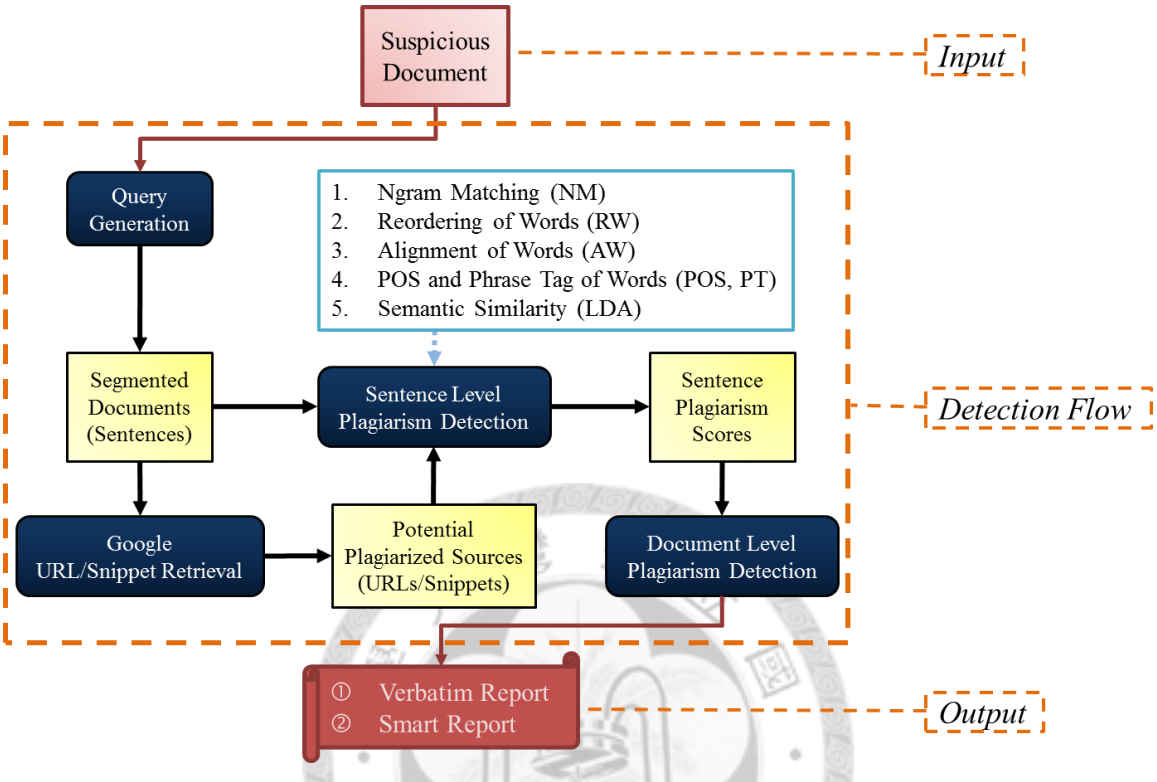

Figure 1. System Architecture

Our system architecture is shown above in Figure 1. Given a suspicious document, we will first segment the document into words and then into sentences. For Chinese input we use CKIP[9] provided by Academia Sinica[10]. Each sentence will be treated as a query and will be sent to the Google[11] search engine twice, quoted and unquoted. We then retrieve the top 30 results returned by the search engine. All of the top 30 returned links of each quoted sentence query will be listed in the report of Verbatim Plagiarism.

For the unquoted sentence queries, the snippets provided in the top 30 results are further processed to detect smart plagiarism. We will first segment each snippet into

[9] CKIP: http://ckipsvr.iis.sinica.edu.tw/

[10] Academia Sinica: http://www.sinica.edu.tw/index.shtml

[11] Google: https://www.google.com/

smaller pieces by "..." because the symbol "..." provided by Google indicates that those pieces are somewhat distant away from each other in the original article. We then cut each piece into sentences. Every sentence in the snippet will be further examined with the corresponding query sentence to perform "Sentence Level Plagiarism Detection", which uses six features: **NM**, **RW**, **AW**, **POS**, **PT**, and **LDA.** The six features will be explained in Section 3.2. Each feature will output a prediction score on whether plagiarism is detected in a given sentence. We used an ensemble method, which is discussed in Section 3.3, to merge the scores from the six features. Given the scores of each sentence provided by the ensemble, the score of the highest-scoring sentence will be added to the score of the link corresponding to the currently processed snippet. Given the accumulated scores for each snippet, we perform "Document-Level Plagiarism Detection" on each suspicious source Web document, or link, with a cutoff threshold, and finally output a ranked list as the report of Smart Plagiarism. The rank in the list as well as the rank score of the link reflects the degree of how likely that it may be a possible plagiarized source. Note that before outputting the report of Smart Plagiarism, we have performed a post proccessing step to filter out the links that are already reported in the verbatim report.

The following sections will explain the aforementioned steps in more detail.

3.1 Query a Search Engine

We first break down each article into a series of queries or sentences to query a search engine. Google is used by default. Several systems such as ([7], Liu et al., 2007) have proposed a similar idea. The main difference between our method and theirs is that we send not only quoted queries but also *unquoted* ones. We do not require the search

results to completely match to the query sentence. This strategy allows us to not only identify the copy/paste type of plagiarism but also re-written/edited type of plagiarism.

3.2 Sentence Level Plagiarism Detection

Since not all outputs of a search engine contain an exact copy of the query, we need a model to quantify how likely each of them is the source of plagiarism. For better efficiency, our experiment exploits the snippet of a search output returned by Google to represent the whole document. That is, we want to measure how likely a snippet is the plagiarized source of the query. We designed several models which utilized rich lexical, syntactic and semantic features to pursue this goal, and the details are discussed below.

3.2.1 Ngram Matching (NM)

One straightforward measure is to exploit the ngram similarity between source and target text. Given two sentences, *S*(source) and *T*(target), we first enumerate all ngrams in *S*, and then calculate the amount of duplication ngrams with those in *T*. The ngram similarity can be measured with three different formulas illustrated below in (1), (2), and (3). Note that our matching is based on stemmed ngrams.

$$\mathrm{NM}_S = \frac{\#\text{ of matched ngrams}}{\#\text{ of ngrams in } S} \qquad (1)$$

$$\mathrm{NM}_T = \frac{\#\text{ of matched ngrams}}{\#\text{ of ngrams in } T} \qquad (2)$$

$$\mathrm{NM}_{\mathrm{avg}} = \frac{\#\text{ of matched ngrams}}{(\#\text{ of ngrams in } S + \#\text{ of ngrams in } T)\ /\ 2} \qquad (3)$$

It seems that NM_{avg}, is a better fit for our need. However, when the length of the source and target are imbalanced, NM_{avg} itself cannot reflect the degree of plagiarism very well. We deal with this issue by considering both NM_S and NM_T with a pre-defined threshold *TH*=0.5. If min (NM_S, NM_T) > *TH*, then the **NM** score will be defined as max (NM_S, NM_T), otherwise it is NM_{avg}.

For the choice of n, the larger n is, the harder for this feature to detect plagiarism with insertion, replacement, and deletion. According to ([4], Cedeno and Rosso, 2009)' s experiments on the METER[12] corpus, their best results are obtained when considering low level word ngrams comparisons (n={2, 3}). And in our experiment on the sampled PAN-2010 corpus in English as well as the annotated Web document dataset in Chinese, which will both be further introduced in section 4.1, we chose n=2.

3.2.2 Reordering of Words (RW)

Plagiarism can come from the reordering of words. We argue that the permutation distance between *S* and *T* is an important indicator for reordered plagiarism. The permutation distance is defined as the minimum number of pair-wise exchange between *matched words* needed to transform a target sentence, *T*, into the same order of matched words as a source sentence, *S*, and Figure 2 below is a simple example.

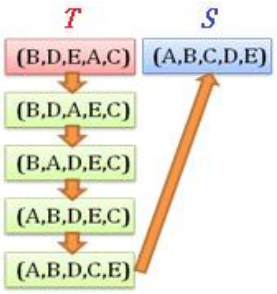

Figure 2. An example of reordering of words

[12] The METER corpus: http://nlp.shef.ac.uk/meter/

As mentioned in ([12], Sörensena and Sevaux, 2005), the permutation distance can be calculated by expressions (4) and (5):

$$d(S,T) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} z_{ij} \qquad (4)$$

where

$$z_{ij} = \begin{cases} 1, & S(\mathrm{i}) > T(\mathrm{j}) \text{ and } S(\mathrm{j}) < T(\mathrm{i}) \\ 0, & otherwise \end{cases} \qquad (5)$$

S(i) and T(i) indicates the i[th] matched word in S and T respectively and n is the number of matched words between them. Let μ be the normalized term, which is the maximum possible distance between S and T, as shown in (7), then the reordering score of the two sentences, expressed as **RW**(S, T), will be (6):

$$\mathbf{RW}(S,T) = 1 - \frac{\mathrm{d}(S,T)}{\mu} \qquad (7)$$

where

$$\mu = \frac{\mathrm{n}^2 - \mathrm{n}}{2} \qquad (6)$$

3.2.3 Alignment of Words (AW)

Besides reordering, plagiarists often insert words into a sentence or delete some from it. We tried to model such behavior by finding the alignment of two word sequences. We performed the alignment using a dynamic programming method as mentioned in ([14], Wagner and Fischer, 1975). As shown in Figure 3, a word match earns 2 points, while a word mismatch receives a penalty of -1 points. A gap also gets a penalty of -1 points. Since each gap may span across more than one word, for each dash symbol("-") in gaps covering a word another -1 points will be added. The alignment algorithm tries to

maximize the score of the summation of each point produced by the different types of matching result.

C	-	A	-	T	A	A	C	T	
C	G	G	A	C	A	-	-	T	
+2	-1	-1	-1	-1	+2	-1	-1	+2	= 0
		Alignment score: 0 -1 -1 -1 = -3							

Figure 3. Alignment of words

However, such alignment score does not reflect the continuity of the matched words, which can be an important cue to identify plagiarism. To overcome such drawback, we revise the score as below.

$$\mathbf{AW} = \frac{\sum_{i=1}^{|M|-1} G_i}{|M|-1} \qquad (8)$$

where

$$G_i = \frac{1}{\#\text{ of words between}(M_i, M_{i+1})+1} \qquad (9)$$

M is the list of matched words, and M_i is the i^{th} matched word in M. This implies we prefer fewer unmatched words in between two matched ones. Consider the following case in Figure 4.

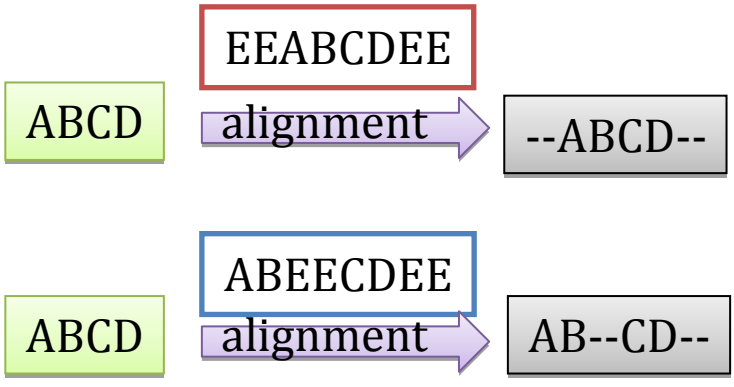

Figure 4: An alignment example

We can tell that when considering aligning "ABCD" with the two patterns, "EEABCDEE" with the alignment result "--ABCD--" should be more continuous than "ABEECDEE" with the alignment result "AB--CD--", but by the alignment algorithm, they will get the same scores. By the redefined way of calculation, the two cases are with the score of 3 and 2.33 respectively, and thus in terms of alignment "--ABCD--" is considered more similar based on this measure. As a result, after the alignment is found, we recalculate the alignment score such that this similarity can be better represented.

3.2.4 POS and Phrase Tag of Words (POS, PT)

Exploiting only lexical features can sometimes result in some false positive cases because two sets of matched words can play different roles in the sentences. *S* and *T* in Table 3 is a possible false positive case:

Table 3. An example of matched words with different POS and phrase tags

S: The man likes the well dressed young woman.				
T: The face of the woman in red dress looks like the man's one.				
Word	*S*: POS	*T*: POS	*S*: PT	*T*: PT
man	*NN*	*NN*	*NP*	*NP*
like	*VBZ*	*IN*	*VP*	*PP*
dress	*JJ*	*NN*	*ADJP*	*NP*
woman	*NN*	*NN*	*NP*	*NP*
POS	*NN*: Noun			
	VBZ: Verb, 3rd person singular present			
	IN: Preposition			
	JJ: Adjective			
PT	*NP*: Noun Phrase			

	VP: Verb Phrase
	PP: Prepositional Phrase
	ADJP: Adjective Phrase

Therefore, we further explore syntactic features for plagiarism detection. To achieve this goal, we utilize the Stanford Parser[13] to obtain POS and phrase tags of the words. For simplicity we abbreviate POS tags as **POS** and phrase tags as **PT**. Then we design an equation to measure the **POS** and **PT** similarity, which is shown below in (10).

$$\textbf{POS/PT} = \frac{\#\text{ of matched words with identical } \textbf{POS/PT}}{\#\text{ of matched words}} \qquad (10)$$

We paid special attention to the case when a sentence is transformed from an active form to a passive-form or vice versa. A subject originally in a Noun Phrase can become a Prepositional Phrase, i.e. "by …", in the passive form while the object in a Verb Phrase can become a new subject in a Noun Phrase. Here we utilize the Stanford Dependency provided by Stanford Parser to match the POS/PT between active and passive sentences. In other words, we handle only 3 kinds of phrase tag : *NP*, *VP*, *PP*. For all other kinds of phrase tags, our system will assign the word with the "ELSE" tag.

3.2.5 Semantic Similarity (LDA)

Plagiarists, sometimes, replace words or phrases with those that contain similar meanings. While previous works ([6], Li et al., 2006) often explore semantic similarity using lexical databases such as WordNet to find synonyms, we exploit a topic model, specifically Latent Dirichlet Allocation ([1], David M. Blei et al., 2003), to extract the semantic features of sentences. Given a set of documents represented by their word

[13] Stanford Parser, a statistical parser: http://nlp.stanford.edu/software/lex-parser.shtml

sequences, and a topic number n, LDA learns the word distribution for each topic and the topic distribution for each document to maximize the likelihood of the word co-occurrence in a document. The topic distribution is often taken as the semantics of a document. We use LDA to obtain the topic distribution of a query and a candidate snippet, and compare the cosine similarity of them as a measure of their semantic similarity. To handle the case that words in the source sentence may be reordered, we have tried another approach by calculating the overlap percentage of LDA tags as the **LDA** score. The computing details are the same as those illustrated in calculating the **NM** score. According to our experiment, the latter approach does perform better.

The details of the training data used to train the LDA models are as follows. For English training data, we use the PAN-2010 Corpus. For Chinese training data, we retrieved 85 review articles from the Web randomly, where 33 of them are book reviews, 32 of them are movie reviews and the rest 20 of them are reviews of music albums.

3.3 Ensemble Similarity Scores

Up to this point, for each snippet the system generates six similarity scores to measure the degree of plagiarism in different aspects. In this stage, we propose two strategies to linearly combine the scores to make better prediction. The first strategy utilizes each model's predictability (e.g. AUC) as the weight to linearly combine the scores. In other words, the models that perform better individually will obtain higher weights. In the second strategy we exploit a learning model (in the experiment we use Liblinear[14]) to learn the weights directly.

[14] Liblinear: http://www.csie.ntu.edu.tw/~cjlin/liblinear/

3.4 Document Level Plagiarism Detection

For each query from the input article, our system assigns a degree-of-plagiarism score to candidate URLs which could be the source of plagiarism. In order to clearly represent the degree of plagiarism for each candidate, we aim to give a ranked list of all plagiarized source candidates. Each candidate contains a certain number of sentences, and the ensembled score of each sentence is computed as described in Section 3.3. We merge the scores of each sentence to derive the score of each candidate using the equation shown in (11).

$$\sum_{i=1}^{\#\text{ of sentences associated with the candidate}} S_i \qquad (11)\ ,$$

where

$$S_i = \begin{cases} E_i, & E_i \geq 0.5 \\ 0, & E_i < 0.5 \end{cases}, \quad E_i\text{: Ensembled score of query sentence i} \qquad (12)$$

We set up a cutoff threshold, 0.5, to obtain the most plausible URLs. At the end, the candidates are sorted by their scores to produce a ranked list, and our system highlights the suspicious areas of plagiarism for display.

Chapter 4 Evaluation

We evaluated our system from two different perspectives. We first evaluated the *sentence level plagiarism detection* using the English PAN-2010[15] corpus. We then evaluated the capability of the full system to detect on-line plagiarism using our own Chinese dataset which was crawled from the Web.

4.1 Dataset

In this section, we will give a detailed illustration of the two datasets we used for the corresponding evaluation tasks.

4.1.1 PAN-2010 Corpus

To compare the detection capability of our model with the state-of-the-art methods, we need a well-known dataset that researchers in this area would use and test their algorithms on. The International Competition on Plagiarism Detection is a large tournament held by PAN since 2009. Every year more than ten research groups from various countries take part in it. The corpus for the competition apparently meets our need.

However, the competition in PAN is designed for off-line plagiarism detection; the competitors does not exploit an IR system to search the Web like we do. Nevertheless, we can still compare the core component of our system, the sentence-based measuring model, with that of other systems. To achieve this goal as well as complete the detection process in a shorter and a more reasonable time, we first randomly sampled 370

[15] PAN, which is abbreviated from International Workshop on **P**lagiarism Analysis, **A**uthorship Identification, and **N**ear-Duplicate Detection. Website of PAN-2010 can be found at: http://pan.webis.de/

documents from PAN-2010 external plagiarism corpus ([11], Martin Potthast et al., 2010) which contains 2882 labeled plagiarism cases. The distribution of our sampled dataset is the same as the original PAN-2010 corpus, which consists of 50% source documents, and 50% suspicious documents with half containing plagiarism cases while the other half do not. We exclude both the translated plagiarism and simulated plagiarism in the external plagiarism corpus. The former one is dismissed for that it is out of our focus. The latter one is also excluded because even by manual checking we can hardly find a sign of simulated plagiarism in many plagiarized cases in the golden standard.

4.1.2 Chinese Web Documents

To evaluate the overall system, we need some real-world plagiarism cases in the WWW. There is a large variety of real-world plagiarism cases in the WWW, but it is difficult to deal with all cases at the same time. Therefore, we only focus on review articles of books, movies and music albums. We manually collected 60 real-world review articles from the Internet for books (20), movies (20), and music albums (20). Details of the review articles including titles as well as the source links can be found in Appendix 1.

However, for an online system like ours, there is no ground truth available to perform system evaluation. To overcome such a difficult situation, we manually annotated the ground truth. We first randomly chose 30 out of the 60 reviews, 10 for each category. Then we broke each of the review documents into sentences and used the sentences as queries to Google. We retrieved 5636 pieces of snippet candidates in total.

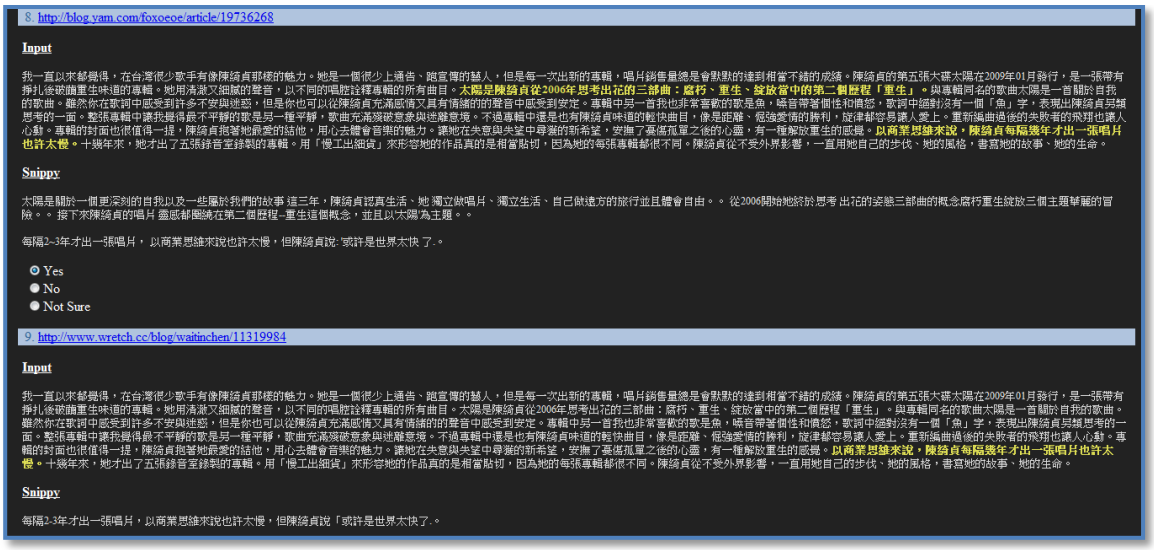

Figure 5: A snapshot of the annotation system

In order to annotate the snippet candidates, we built an annotation system as shown above in Figure 5. We asked 63 people to annotate whether those snippets represent plagiarism cases of the original review article, and have each snippet pair be annotated at least twice by diffent annotators. To unify the judging standard, we had told all the annotators several criteria before they started their annotation. The criteria are listed below in Table 4.

Table 4. Criteria for the annotators.

1.	Focus on those query sentences marked in yellow and words near them. Compare the corresponding area with the snippet below, if any part of them can be considered as a suspicious case, annotate it as a positive plagiarism case.
2.	In the input article, if there exists one sentence which is longer than 10 words and that it matches a certain part of the snippet entirely, annotate it as a positive plagiarism case.
3.	If the snippet is empty, annotate it as a negative case.

Eventually we have obtained an annotated dataset and found a total of 502 plagiarized candidates with 4966 non-plagiarized candidates for evaluation, which implies that our assumption is not totally unfounded.

4.2 Sentence-based Evaluations on PAN-2010

We compared the performance of our system and existing systems in the sentence-based plagiarism detection task. Given a suspicious passage and a set of snippets which contains the source of the suspicious passage, we would like each system to return a ranked list of snippets, where the snippet with the highest rank is the most probable source of the suspicious passage. The system with the best ranked list is the best system. In order to obtain high-quality negative examples for the set of snippets for evaluation, we built a full-text index on our sampled PAN-2010 corpus using the Lucene package. Then we use the suspicious passages as queries to search the whole dataset using Lucene. Since there is a length limitation in Lucene (as well as in the real-world search engines), we further broke the 2882 plagiarism cases into 6477 queries. We then extracted the top 30 snippets returned by Lucene as the potential negative candidates for each plagiarism case. Note that for each suspicious passage, there is only one target passage (given by the ground truth) that is considered as a positive plagiarism case in this data, and it can be either among these 30 snippets or not. However, we combined the 30 snippets with the ground truth, and used our (as well as the competitors') models to rank the degree-of-plagiarism for all the candidates. We then evaluated the rank by the area-under-PR-curve (AUC) score. We compared our system with the winning entry of PAN-2011 ([5], Grman and Ravas, 2011) and the stopword ngram model by ([13], Stamatatos, 2011) that claims to perform better than the winning entry. The results of each individual model and ensemble using 5-fold cross validation are listed in Table 5. It shows that NM is the best individual model, and an ensemble of three features outperforms the state-of-the-art by 26%.

Table 5. Sentence-based evaluations.

(a) AUC of single models; (b) AUC of other state-of-the-art algorithms and ours

NM	*RW*	*AW*	*PT*	*PP*	*LDA*
0.876	0.596	0.537	0.551	0.521	0.596

(a)

	Ours ensemble	*PAN-2011 Champion*	*Stopword Ngram*
AUC	0.882 (NM+RW+PP)	0.620	0.596

(b)

4.3 Full System Evaluations on Chinese Web Documents

To evaluate performance on sentence level plagiarism, we used the annotated dataset we built manually by human annotators, as described in Section 4.1.2. Table 6 shows the average AUC of 5-fold cross validation. The results show that our method outperforms the PAN-2011 winner slightly, and is much better than the Stopword Ngram.

Table 6. Full system evaluations on Chinese Web documents.

(a) AUC of single models; (b) AUC of other state-of-the-art algorithms and ours

NM	*RW*	*AW*	*PT*	*PP*	*LDA*
0.904	0.778	0.874	0.734	0.622	0.581

(a)

	Ours ensemble	*PAN-2011 Champion*	*Stopword Ngram*
AUC	0.919 (NM+RW+AW+PT+PP+LDA)	0.893	0.568

(b)

4.4 Discussion

There is some inconsistency of the performance of single features in the two experiments performed on the PAN-2010 data set and the Chinese Web Document data

set. The main reason we believe is that the plagiarism cases were created in very different manners. Plagiarism cases in PAN external source are created artificially through word insertions, deletions, reordering and synonym substitutions. As a result, features such as word alignment and reordering did not perform well because they did not consider the existence of synonym word replacement. On the other hand, real-world plagiarism cases returned by Google are those with matching-words, and we can find better performance for *AW*.

The performances of syntactic and semantic features, namely *PT, PP and LDA*, are consistently inferior than other features. It is because they often introduce false-positives as there are some non-plagiarism cases that might have highly overlap syntactic or semantic tags. Nevertheless, experiments also show that these features can improve the overall accuracy in the ensemble.

We also found that the *stopword ngram* model is not applicable universally. For one thing, it is less suitable for on-line plagiarism detection, as the length limitation for queries diminishes the usability of stopword ngrams. For another, Chinese seems to be a language that does not rely as much on stopwords as the Latin languages do to maintain its syntax structure.

Samples of our system’ s finding can be found here, http://tinyurl.com/6pnhurz.

Chapter 5 System Demonstration

We developed an online demo system using JAVA (JDK 1.7) and GibbsLDA++[16]. The system currently supports the detection of documents in both English and Chinese. Our online system can be found here: http://mslab.csie.ntu.edu.tw/~ubiquitin/opd/detect.php

Users can either upload the plain text file of a suspicious document, or copy/paste the content onto the text area, as shown below in Figure 6.

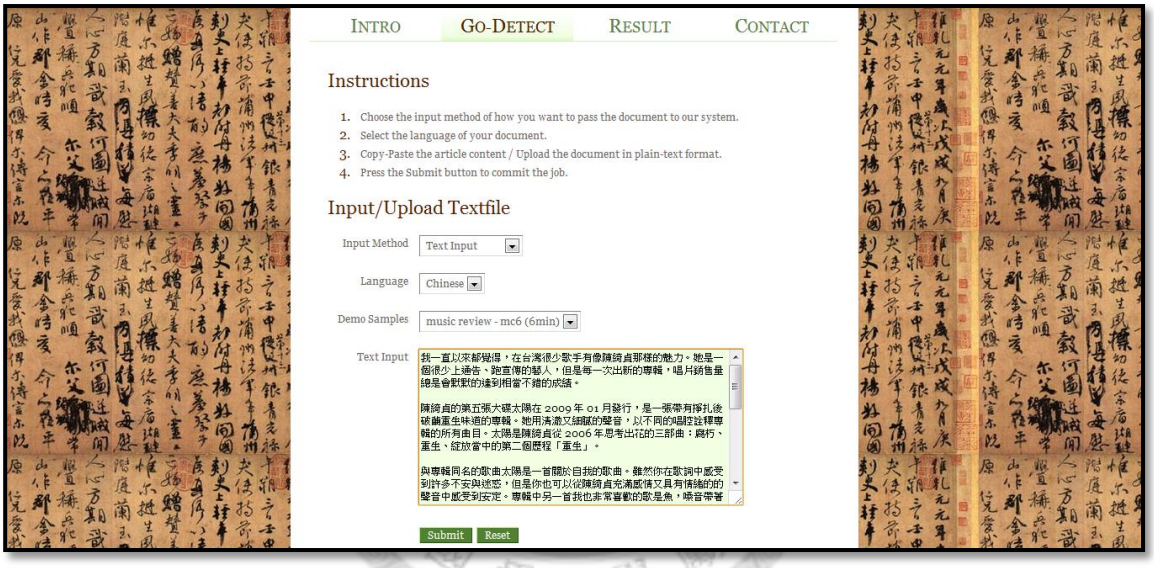

Figure 6. An overview of our *Text Input* interface

When the suspicious document is ready, the user can press the submit button to start the detection process. The input content will firstly be emitted onto the top of the screen followed with the estimated processing time by our program. Such information is provided to prevent users from closing the page before the detection is finished after waiting too long.

It takes around 5-10 seconds in average for the system to process an English sentence, and 10-15 seconds for a Chinese sentence. The bottleneck lies mainly in Stanford-Parser-

[16] GibbsLDA++: http://gibbslda.sourceforge.net/

Tagging and LDA-Tagging. For Chinese inputs, the segmentation of words through Web query from either Y! 斷章取義 or CKIP also takes a relatively long time.

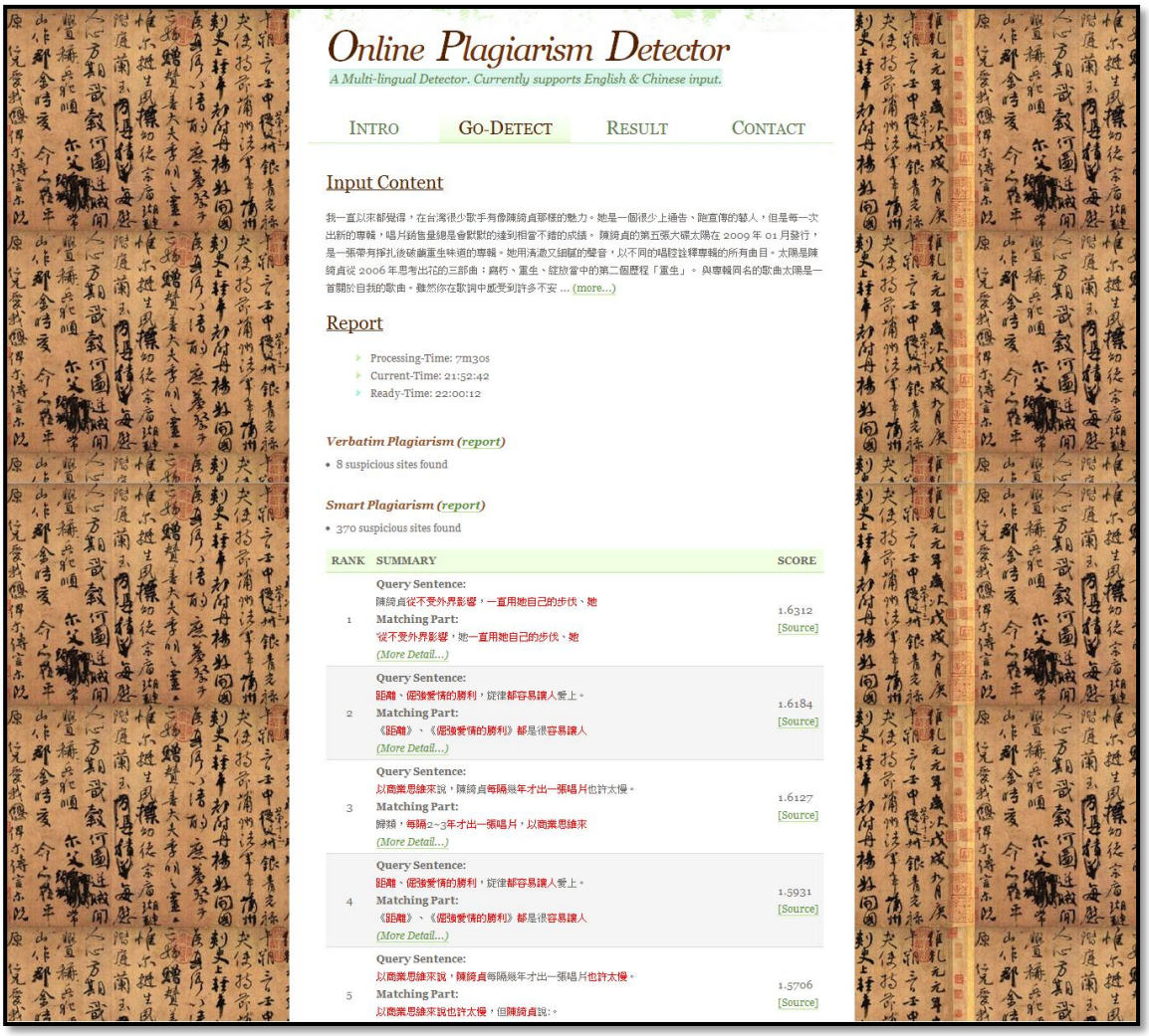

Figure 7. An overview of the outputs

After the system processing is over, the user can see a clear overview of the detection results shown above in Figure 7. Since the verbatim cases are all copy/paste type of plagiarism, which is easy to detect and has no need for a second-pass check by user, our system will not print out the report to overwhelm the page.

On the other hand, the report of smart plagiarism may contain suspicious online articles which have similar syntax structures or semantic meanings but are not entirely

identical to the input document. Therefore, we print out the summary report displaying some URLs and snippets as the potential source of plagiarism. Each row is a suspicious Web document. We print the highest-scored sentence pair in brief as a representative, and for more suspicious sentence pairs found in the same document, the user can click the "More Details" hyperlink, which is also provided in the report under it.

Figure 8 is a sample detail view of a suspicious case of verbatim plagiarism, while Figure 9 is a sample of smart plagiarism.

SUSPICIOUS

1. Also appears in other **4** pages

Overlap Sentence	我一直以來都覺得，在台灣很少歌手有像陳綺貞那樣的魅力。
Total Score	1.0

2. Also appears in other **3** pages

Overlap Sentence	她是一個很少上通告、跑宣傳的藝人，但是每一次
Total Score	1.0

3. Also appears in other **3** pages

Overlap Sentence	出新的專輯，唱片銷售量總是會默默的達到相當不錯的成績。
Total Score	1.0

4. Also appears in other **1** pages

Overlap Sentence	陳綺貞的第五張大碟太陽在2009年01月發行

Figure 8. Detail view of a suspicious case of verbatim plagiarism

SUSPICIOUS

1. Also appears in other **95799** pages

Query Sentence	陳綺貞從不受外界影響，一直用她自己的步伐、她
Matching Part	'從不受外界影響，她一直用她自己的步伐、她
Total Score	0.8627

2. Also appears in other **13899** pages

Query Sentence	太陽是陳綺貞從2006年思考出花的三部曲：
Matching Part	從2006開始她終於思考出花的姿態三部曲的概念。
Total Score	0.7225

3. Also appears in other **5969** pages

Query Sentence	以商業思維來說，陳綺貞每隔幾年才出一張唱片也許太慢。
Matching Part	以商業思維來說也許太慢，但陳綺貞說:'或許是世界太快。
Total Score	0.717

Figure 9. Detail view of a suspicious case of smart plagiarism

Chapter 6 Conclusion

We provide a solution for online plagiarism detection. Comparing to other online plagiarism detection systems, ours exploit more sophisticated features by modeling how human beings plagiarize online sources. We have exploited sentence level plagiarism detection on lexical, syntactic and semantic levels. It can detect not only verbatim copy, but also those articles on the WWW with different degree of common modification techniques performed, such as merging sentences, phrase substitution or reordering, word insertion or deletion, and even paraphrasing, by clever plagiarizers. Another noticeable fact is that our approach is almost language independent. Given a parser and a POS tagger of a language, our framework can be extended to support plagiarism detection for that language. Experiments performed on English and Chinese corpora demonstrate that our system can not only find considerable amount of real-world online plagiarism cases but also outperforms several state-of-the-art algorithms.

References

[1] David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. 2003. Latent Dirichlet Allocation. Journal of Machine Learning Research, 3:2003.

[2] Bear F. Braumoeller and Brian J. Gaines. 2001. Actions Do Speak Louder Than Words: Deterring Plagiarism with the Use of Plagiarism-Detection Software. In Political Science & Politics, 34(4):835-839.

[3] Sergey Brin, James Davis, and Hector Garcia-molina. 1995. Copy Detection Mechanisms for Digital Documents. In Proceedings of the ACM SIGMOD Annual Conference, 24(2):398-409.

[4] Alberto Barrón Cedeño and Paolo Rosso. 2009. On Automatic Plagiarism Detection based on n-grams Comparison. In Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval, ECIR 2009, LNCS 5478:696-700, Springer-Verlag, and Berlin Heidelberg,

[5] Jan Grman and Rudolf Ravas. 2011. Improved implementation for finding text similarities in large collections of data. In Proceedings of PAN 2011.

[6] Yuhua Li, David McLean, Zuhair A. Bandar, James D. O' Shea, and Keeley Crockett. 2006. Sentence Similarity Based on Semantic Nets and Corpus Statistics. In Proceedings of the IEEE Transactions on Knowledge and Data Engineering, 18(8):1138-1150.

[7] Yi-Ting Liu, Heng-Rui Zhang, Tai-Wei Chen, and Wei-Guang Teng. 2007. Extending Web Search for Online Plagiarism Detection. In Proceedings of the IEEE International Conference on Information Reuse and Integration, IRI 2007.

[8] Wan-Yu Lin, Nanyun Peng, Chun-Chao Yen, and Shou-de Lin. 2012. Online Plagiarized Detection Through Exploiting Lexical, Syntax, and Semantic Information. In Proceedings of ACL 2012 Demo.

[9] Sebastian Niezgoda and Thomas P. Way. 2006. SNITCH: A Software Tool for Detecting Cut and Paste Plagiarism. In Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education, p.51-55.

[10] Maria Soledad Pera and Yiu-kai Ng. 2010. IOS Press SimPaD: A Word-Similarity Sentence-Based Plagiarism Detection Tool on Web Documents. In Journal on Web Intelligence and Agent Systems, 9(1).

[11] Martin Potthast, Benno Stein, Alberto Barrón Cedeño, and Paolo Rosso. 2010. An Evaluation Framework for Plagiarism Detection. In 23rd International Conference on Computational Linguistics (COLING 10). Association for Computational Linguistics.

[12] Kenneth Sörensena and Marc Sevaux. 2005. Permutation Distance Measures for Memetic Algorithms with Population Management. In Proceedings of 6th Metaheuristics International Conference.

[13] Efstathios Stamatatos, "Plagiarism Detection Based on Structural Information" in Proceedings of the 20th ACM international conference on Information and knowledge management, CIKM'11

[14] Robert A. Wagner and Michael J. Fischer. 1975. The String-to-string correction problem. In Journal of the ACM, 21(1):168-173.

[15] Daniel R. White and Mike S. Joy. 2004. Sentence-Based Natural Language Plagiarism Detection. In Journal on Educational Resources in Computing JERIC Homepage archive, 4(4).

[16] Du Zou, Wei-jiang Long, and Zhang Ling. 2010. A Cluster-Based Plagiarism Detection Method. In Lab Report for PAN at CLEF 2010.

Appendix

Type		Title	Source Link
Reviews of Books	bc1	默默地我相信天使	http://www.yumau.com/reading/art/5059
	bc2	落花流水	http://www.yumau.com/reading/art/88
	bc3	追風箏的孩子	http://www.yumau.com/reading/art/5183
	bc4	如此蒼白的心	http://www.yumau.com/reading/art/2912
	bc5	難以承受的告別	http://www.yumau.com/reading/art/5088
	bc6	偷書賊	http://www.yumau.com/reading/art/2759
	bc9	殘酷天才	http://www.yumau.com/reading/art/5074
	bc10	網球鞋女孩	http://www.yumau.com/reading/art/5126
	bc11	喀布爾的書商，和他的女人	http://www.yumau.com/reading/art/2898
	bc12	默默地我相信天使(2)	http://www.yumau.com/reading/art/5098
	bc13	博士熱愛的算式	http://www.yumau.com/reading/art/543
	bc14	芬蘭驚豔	http://www.yumau.com/reading/art/2858
	bc15	丁莊夢	http://www.yumau.com/reading/art/43
	bc16	來不及穿的8號鞋	http://www.yumau.com/reading/art/2773
	bc17	肅清之門	http://www.yumau.com/reading/art/2790
	bc18	武則天	http://www.yumau.com/reading/art/2839
	bc19	維納斯的誕生	http://www.yumau.com/reading/art/2806
	bc20	蝴蝶春夢	http://www.yumau.com/reading/art/95
Reviews of Movies	c1	放牛班的春天	http://blog.yam.com/ncculib
	c2	神火之賊	http://www.im.tv/Blog/3023856/1080926
	c3	歡樂谷	http://blog.yam.com/ncculib
	c4	佐賀的超級阿嬤	http://blog.yam.com/ncculib

Reviews of Music Album	c6	天堂的孩子	http://blog.yam.com/ncculib
	c7	艾蜜莉的異想世界	http://blog.yam.com/ncculib
	c8	嘰哩咕與女巫	http://blog.yam.com/ncculib
	c12	益智風雲	http://blog.yam.com/ncculib
	c13	瓦力	http://app.atmovies.com.tw/eweekly/eweekly.cfm?action=edata&vol=178&eid=v178108
	c14	荷頓奇遇記	http://app.atmovies.com.tw/eweekly/eweekly.cfm?action=edata&vol=162&eid=v162108
	c15	長江七號	http://app.atmovies.com.tw/eweekly/eweekly.cfm?action=edata&vol=154&eid=v154111
	c16	瘋狂理髮師	http://app.atmovies.com.tw/eweekly/eweekly.cfm?action=edata&vol=152&eid=v152110
	c17	刺殺傑西	http://app.atmovies.com.tw/eweekly/eweekly.cfm?action=edata&vol=149&eid=v149109
	c18	刺殺傑西	http://app.atmovies.com.tw/eweekly/eweekly.cfm?action=edata&vol=149&eid=v149109
	c19	太陽浩劫	http://app.atmovies.com.tw/eweekly/eweekly.cfm?action=edata&vol=112&eid=1112007
	c20	K歌情人	http://app.atmovies.com.tw/eweekly/eweekly.cfm?action=edata&vol=104&eid=1104008
	mc1	謝安琪 <SLOWNESS>	http://wp.plem.com/?p=6426
	mc2	陳亦迅 <上五樓的快活>	http://wp.plem.com/?p=6040
	mc3	張惠妹 <阿密特>	http://wp.plem.com/?p=4345
	mc4	蔡依琳 <花蝴蝶>	http://wp.plem.com/?p=3532
	mc5	謝安琪<YELLING>	http://wp.plem.com/?p=3530
	mc6	陳綺真 <太陽>	http://wp.plem.com/?p=2015
	mc7	陳姍妮 <回歸本質陳姍妮>	http://wp.plem.com/?p=1332
	mc8	許哲佩 <雪人>	http://3cmusic.com/

	mc9	黃小琥 <沒那麼感人>	http://3cmusic.com/
	mc10	張懸 <南國的孩子>	http://tw.myblog.yahoo.com/music-player/article?mid=23276&prev=23695&next=23207&l=f&fid=6
	mc11	謝和弦 <雖然很芭樂>	http://tw.myblog.yahoo.com/music-player/article?mid=23207&prev=23276&next=21368&l=f&fid=6
	mc12	蕭敬騰 <王妃>	http://tw.myblog.yahoo.com/music-player/article?mid=23695&next=23276&l=f&fid=6
	mc13	林宇中 <淋雨中>	http://tw.myblog.yahoo.com/music-player/article?mid=18962&prev=21368&next=18162&l=f&fid=6
	mc14	凡人 <凡人和他的朋友>	http://tw.myblog.yahoo.com/music-player/article?mid=16806&prev=18162&next=15747&l=f&fid=6
	mc15	楊培安	http://tw.myblog.yahoo.com/music-player/article?mid=13813&next=13584&l=f&fid=6
	mc16	劉若英	http://tw.myblog.yahoo.com/music-player/article?mid=13584&prev=13813&next=13228&l=f&fid=6
	mc17	蔡純佳 <慶幸擁有蔡純佳>	http://tw.myblog.yahoo.com/music-player/article?mid=13228&prev=13584&next=13098&l=f&fid=6
	mc18	王宛之 <我真的受傷了>	http://tw.myblog.yahoo.com/music-player/article?mid=13098&prev=13228&next=13003&l=f&fid=6
	mc19	潘瑋柏 <玩酷>	http://tw.myblog.yahoo.com/music-player/article?mid=11877&prev=13003&next=11054&l=f&fid=6
	mc20	溫嵐 <熱浪>	http://tw.myblog.yahoo.com/music-player/article?mid=11054&prev=11877&next=23587&l=f&fid=6