Automatic Extraction of Publication Time from News Search Results

Yiyao Lu, Weiyi Meng, Wanjing Zhang
Department of Computer Science
SUNY at Binghamton
Binghamton, NY 13902, USA
{ylu0, meng, wzhang5}@binghamton.edu

King-Lup Liu
Webscalers, LLC
Lafayette, LA 70506, USA
kliu@webscalers.com

Clement Yu
Dept of Computer Science
U. of Illinois at Chicago
Chicago, IL 60607, USA
yu@cs.uic.edu

Abstract

The publication time of a page can have a big impact on its relevance to a query, especially for time-sensitive pages such as news items. For news search engines, the publication time of news items can usually be found in the returned search result records. In this paper, we introduce a method that can automatically extract the publication time for each news story returned from news search engines based on several important observations we made. We also introduce a wrapper implementation for the extraction method. The experimental results using data collected from 50 news search engine show that our method is effective and the wrapper implementation can not only improve the extraction accuracy but also the extraction efficiency.

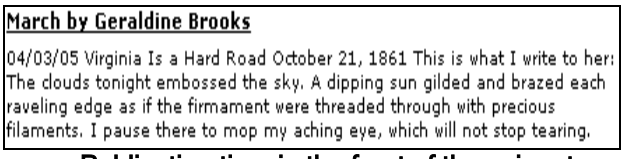
1. Introduction

A search result record (SRR) returned from a news search engine usually contains several pieces of information. In addition to the title, URL, and a short summary (called snippet) of the news article, the publication time of the SRR is also commonly displayed. The publication time can play an important role in several applications. For example, in news event detection and tracking systems, the publication time of a news story is usually one of the most important indicators that help determine whether two news stories refer to the same event [1]. As another example, in news meta-searching systems where the news articles retrieved from multiple sources need to be reorganized into a single ranked list, the publication time is a critical factor used to rank those news articles. In [2], one proposed ranking scheme is to rank the more recent news items higher when multiple news items have the same similarity with respect to a given query. In news metasearch engine AllInOneNews (www.allinonenews.com), the publication time of news items retrieved from different news sources is also used to merge the search results. In this paper, we investigate how to automatically and accurately extract the publication time from the SRRs returned from news search engines.

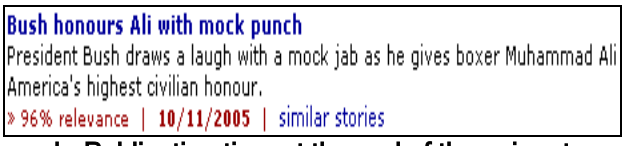
There are many reasons why the problem is a difficult problem. The first is the following heterogeneities among the news sources in displaying the publication time:

1. Different news sites may choose different locations in the result record to display/encode the publication time information. Some prefer putting it close together with the article title while others like to put it in the snippet or even in the URL. Figure 1 shows some examples of publication times at different locations in the SRRs. Among the 50 news sites we collected in our testbed to be used for experiments (see Section 6), 8 search engines display the publication time in title, 36 in snippet, 4 in URL, and 2 in more than one location.
2. Publication times generated by different news sources may consist of different time components. A complete time may consist of year, month, day in a month, hour, minute, second, AM/PM, and even the time zone, but different news sources often adopt different subsets of these components in their publication times. For example, in Figure 1-d, the publication time contains both date and the time in a day while in Figure 1-c, only the date information is provided.
3. Different news sites may represent the same time component differently. For example, in Figure 1-a, the month is represented in digit format, and in 1-c, it is represented in the full word format. Furthermore, some sites may use the abbreviated letter format (e.g. “Oct”) to represent month.
4. Different news sites may give different interpretations to the same time expression. The ambiguity might be caused by the difference in

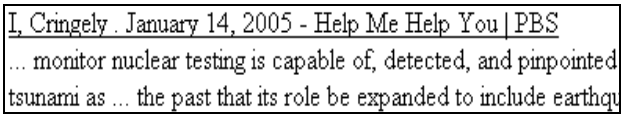
conventions of different countries. For instance, Figure 1-b is a result record returned from *BBC News*, a news source in UK. The expression "10/11/2005" should be interpreted as "November 10th, 2005". But in the US, most likely it will be interpreted as "October 11th, 2005". Even in the same country, the equivocal interpretation problem may still exist because of the habits of different editors. For example, in Figure 1-a, the publication time "April 3rd, 2005" is represented as the expression "04/03/05". But it may refer to "March 5th, 2004" or "March 4th, 2005" to some people.



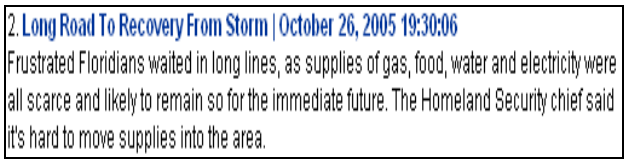
a. Publication time in the front of the snippet



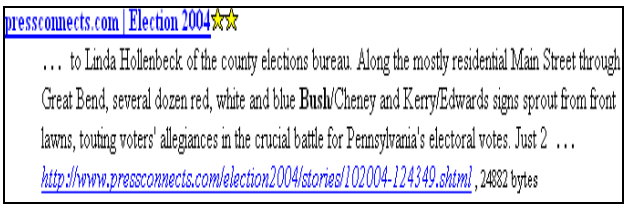
b. Publication time at the end of the snippet



c. Publication time in the front of the title



d. Publication time at the end of the title



e. Publication time in URL

Figure 1. Examples of publication time locations

Another key reason that makes it difficult to correctly extract the publication time is that not every time occurrence corresponds to a publication time. A SRR may contain multiple time occurrences and often only one of them is the correct publication time while others refer to the times of some events. The example depicted in Figure 2 illustrates this situation. We can find multiple temporal information in the title ("March 23, 1998"), snippet ("April 14, 1998", "7:02 PM 3/23/1998", etc), and even in the URL ("980324"), but only the expression at the end of the snippet "04/13/98" is the real publication time of this news article. How to differentiate publication time from non-publication time is another challenge.

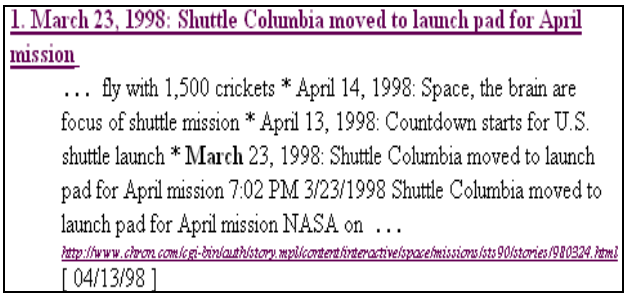


Figure 2. SRR with multiple time information

In this paper, we present a technique for automatically extracting the publication time that appears in the SRRs returned by news search engines. We are not aware of any published work that addresses this issue. This paper has the following contributions. First, we propose a two-step technique to tackle the publication time extraction problem. In the first step, all times are extracted. In the second step, correct publication times are identified. Our method to differentiate publication times from non-publication times takes all the SRRs on the same result page into consideration. Second, we automatically generate a publication time extraction wrapper (extraction rules) for each news site such that the wrapper can be applied to extracting the publication times from the SRRs of new result pages from the same news site. As the wrapper is generated based on the SRRs of multiple training result pages, it captures the publication time pattern(s) more precisely compared to using just one result page. As a result, the use of wrapper can improve extraction accuracy. Moreover, as wrappers are built in advance, they can be readily used to extract publication times from newly returned SRRs, leading to reduced extraction time. Third, experimental results are reported to show that our technique is very accurate.

The rest of the paper is organized as follows. In Section 2, we review some related work. In Section 3, the method used to extract potential publication times is introduced. In Section 4, we present our technique to identify the correct publication time for each search result record. In Section 5, we present a wrapper implementation to improve the extraction efficiency and quality. Section 6 reports our experimental results and

Section 7 concludes the paper.

2. Related work

The temporal information extraction problem has appeared in the information extraction research for a long time (for example, [4, 5]). In previous researches, the extraction is based on the rules for the time expressions. A rule is a description on what time components are used to form a time, the format of each component, and how they are organized. Once the rules are generated, they are used to map the target text strings. If a match can be found, the time can be parsed. The rules they generated, however, are far from enough to represent all time expressions that can be found in the real applications involving a large number of autonomous sources. In TIDES [6], the temporal guidelines with form "YYMMDDhhmmss" based on the Georgian Calendar scheme were used to represent the time information. The "Dates and times" extraction module in [3] generates different rules representing the "plain dates" and the times. But the formats of "plain date" can only be (1) a month followed by a day and a year, (2) a month followed by a year, (3) a month followed by a day, and (4) a day followed by a month and an optional year. Moreover, the month can only be in the word format. The rules such as the combination of numeric month and year are eliminated because they can easily lead to ambiguous interpretations. For this reason, the extraction results of these works are not very satisfactory.

Our work in this paper differs from the previous ones in several aspects. First, the applications involved are different. In our case, extracted publication times are used to rank/merge search results from news search engines. Second, we need to deal with a large variety of time formats because news sites are highly autonomous in selecting the format they use. Some time occurrences appearing in the URLs of SRRs often have unusual formats. We also need to handle possibly different interpretations to the same time expression. Third, we need to differentiate publication times from non-publication times because not all times are publication times.

3. Time extraction

Given a collection of search result records, the first step is to extract all the potential publication times from each record. We have the following observation based on our analysis of sample news sites:

Observation 1. Unlike "yesterday" or "two weeks ago" which represent a relative time point, the publication time usually indicates an absolute time point with the format containing the components including year, month, day, hour, minute, second, am/pm marker, and time zone.

Among the above components, year, month and day are related to the date, and the rest are related to the time in the day.

Whether a term sequence can be parsed into a time is determined by a set of pre-defined rules. For example, if a rule says "a date is a month in text form followed by a day in number form and a year in 4-digit number form", the term sequence "October 29, 2005" will satisfy the rule and will be parsed into a date. In order to express the rules in a systematic way, we define the letter representations for each component (see Table 1). Considering that a component may have text or number presentation formats, we adopt a different representation for each format. Based on this mechanism, the above example rule can be generalized into "MMM dd yyyy". We call each such generalized rule as a *pattern*.

While some news sites provide both date and time, many display one of them, usually the publication date only. In this work, we first extract publication date and the time in the day separately, and then try to merge them together if both are available in the same SRR and appear adjacently. Correspondingly, two separate sets of rules are generated for the extraction of date and time in the day. Figure 3 shows a sketch of our algorithm for extracting all potential publication times (time and date) in a given SRR.

Table 1. Time component pattern representation

Letter Rep.	Time Component	Presentation Format	Example
yy	Year	2-digit number	05
yyyy	Year	4-digit number	2005
MMM	Month in year	Text	Jan, January
MM	Month in year	Number	1
dd	Day in month	Number	31, 31st
hh	Hour in day	Number	10
mm	Minute in hour	Number	59
ss	Second in minute	Number	55
a	am/pm	Text	AM, p.m.
z	Time zone	Text	EST, GMT-08:00

When considering date extraction, the title, snippet and the URL of the input SRR are considered as three target documents (see line 1 in Extract(SRR, patterns) of

Figure 3). For each target document, the terms are processed one by one to check whether it is a date component candidate. If it is, we use the corresponding predefined date patterns to map this term and the subsequent consecutive terms starting from this term. If a pattern matches, the date can then be parsed with this pattern and is saved. During the matching process, we try to match the time pattern with the most number of components first. Once it matches successfully, the shorter patterns that appear as a part of the matched one will not be used. For example, if the pattern "yyyy MMM dd" matches, it is unnecessary to use the pattern "MMM dd" to test against the same term sequence.

```
DateTimeExtractor(SRR)
1    dates ← Extract(SRR, date pattern);
2    times ← Extract(SRR, time pattern);
3    merge dates and times;

Extract(SRR, patterns)
1    for each target document s ∈ SRR
2        for each term t ∈ s
3            if t is a date component candidate
4                then create new Time Set ts;
5                    for each pattern p
6                        d ← match(p, s starting with t);
7                        if d is valid
8                            then ts = ts ∪ {d};
9                    save ts;
```

Figure 3. Date/Time extraction algorithm

A date component candidate is a term that can potentially be a starting component in a date expression or itself matches any date pattern. A term can be considered as a date component candidate in several cases, and in each case, only the patterns matching this candidate are used to do the parsing. First, it is a text representation of a month in a complete form (e.g. "January") or an abbreviated form followed with or without an optional dot (e.g. "Jan", "Jan."). The patterns starting with "MMM" will match this type of candidate. Second, the term is in an ordinal number format. The patterns matching this type of terms can only be the ones starting with "dd" since usually we do not use ordinal numbers to represent other date components. The third case is that the term is a cardinal number, which might be matched with any pattern (1) represented in multiple terms and starting with "yy", "yyyy", "MM", or "dd" (e.g. yyyy MMM dd), (2) represented in single term and with every component in digit format (e.g. yyyyMMdd). The value constraint and the digit length constraint of the components are used to check against the term before the pattern is applied for mapping. The value constraint defines the valid value range of a component. For example, "MM" is between 1 and 12. Therefore, if the term is "13", the patterns starting with "MM" will not be used. The digit length constraint defines the requirement on the term length for matching certain pattern. For example, in Figure 1-e, the term "102004" in the URL is in the cardinal number format represented by 6 digits. Since we usually use two or four digits to represent year information and one or two digits to represent month and day in a month information, this term cannot be a starting component of any date pattern with multiple terms such as "yy MMM dd", "dd MMM yy", or "MM dd yyyy", etc. The patterns matching this term can only be "MMddyy", "ddMMyy", "yyMMdd", where the year is represented by a 2-digit number; "MMyyyy" or "yyyyMM", where day in month information is not present and a 4-digit year pattern is used. It can be seen that the term with a cardinal number format can be a valid date component candidate only if its length is 1, 2, 4, 6, or 8 and under the condition that the value constraint is satisfied. The constraints are introduced to reduce the number of patterns applied to map the same sequence of terms.

A special issue that needs to be addressed is that a date pattern may not contain all the date components. For example, it is very normal that a publication date does not contain the year information. Figure 4 shows an example of this situation. In this case, we need to determine the year for this date. Our solution is as follows. First, we scan all the dates extracted from the same SRR; if there is one having the same month and the day in month with this date and also having the year information, we then assign this year to it. If no such hint can be found, we then set the year to the current year if the extracted date (with only the month and day components) is the same as or before the current date; otherwise, its year is the previous year (it cannot be a future year for a publication time). For example in Figure 4, the extracted date is "Oct 26" and if the current date is "Nov 2nd, 2005", then 2005 is determined to be the year; if the current date is "March 15, 2005", then we assume the date in this SRR is "Oct 26th, 2004".

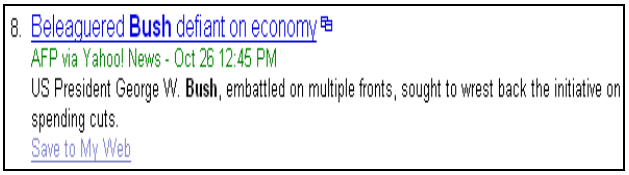


Figure 4. Publication time without year information

For each extracted date, its pattern and the *position*

information are saved. The position information includes the place where the date locates (in the title, snippet, or in the URL), and the distances to the beginning and to the end of the line. The information will be used to identify the correct publication time (see Section 4).

The extraction of the time in a day is similar to the date extraction. Some predefined time patterns, instead of the date patterns, are used. The parsed time, associated with the pattern and the position information, are saved. When all the time information in a result record is extracted, they are merged with the extracted dates. It can be observed that if the publication time contains both date and time in a day, they always locate close to each other. Based on this observation, we only merge the time information with the date that is immediately in front of or after it. Once the date and time are merged, the date pattern and the time pattern are combined into one and the position information is updated accordingly.

While a publication time usually has the date information only, there are occasional cases where the publication time has only time information but no date information (e.g., news from *Yahoo*). This usually happens to very recent news items and the time usually has the following pattern "[xx hours] yy minutes ago" (the hour information may be absent). In this case, the date information and the time in the day can be derived through simple date/time arithmetic (i.e., the current date and time minus the xx hours and yy minutes).

It can be seen that for the same sequence of terms, multiple patterns may be matched and thus multiple valid times may be parsed from them. For example, in Figure 1-a, the expression "04/03/05" may be interpreted as "Apr 3rd, 2005", "Mar 4th, 2005", or "Mar 5th, 2004". This situation usually happens when parsing the date, where multiple date components are represented by cardinal numbers and their values satisfy the constraints for all components. It rarely happens on time expressions. For example, "03:04:05" means "4 minutes and 5 seconds after 3 o'clock" only. It seems that there is a consensus on representing and interpreting the times.

In our approach, the extracted dates are saved as Time Sets rather than individually. A Time Set is a set of times that are parsed from the same sequence of terms. For the above example, "04/03/05", its Time Set is {"Apr 3rd, 2005", "Mar 4th, 2005", "Mar 5th, 2004"}. If the Time Set contains exactly one extracted date, the correct date can be determined easily and we call such a Time Set an *Undisputed Time Set*. The Time Set with multiple dates in it is called *Disputed Time Set*. The times in a Disputed Time Set share the same position information. Clearly, there is only one correct date in a Disputed Time Set and the correct one needs to be identified. We will delay resolving this issue till the publication time identification step to be discussed in Section 4.

4. Publication time identification

In this section, we discuss how to identify the correct publication time from the extracted times for each SRR. We first remove times that are obviously non-publication times such as dates referring to a very old time or some time in the future. Thus, for each result record, only potentially correct publication times remain. These times are stored as units of Time Sets and distributed at different locations within the record. The algorithm for the remaining steps is shown in Figure 5. The main idea of our approach is to consider the extracted times from all the records on the same result page collectively and the details of this algorithm are explained below.

```
PubTimeIdentify
1    for each SRR
2      /* group the Time Sets (TSs) by location */
3      for each location
4          if TITLE-FRONT or SNIPPET-FRONT
5             then keep the very first TS;
6          else if TITLE-END or SNIPPET-END
7             then keep the very last TS;
8          else //URL
9             merge all TSs;
10   compute frequency of each location;
11   find the location L with the max frequency MF;
12   if MF > threshold
13      PTLoc ← L;
14   /* create undisputed time pattern set (UTP) */
15   for each SRR
16      if only one time T exists in the TS at PTLoc
17         then set T as the publication time;
18              add pattern of T to UTP;
19   for each SRR with undetermined publication
     time
20      if time T at PTLoc matches a pattern in UTP
21         then set T as the publication time;
22         else make the selection based on
23              additional information such as
24              the country of the site;
```

Figure 5. Publication time identification algorithm

First of all, we group all extracted Time Sets from each result record into five *locations* (lines 1-9): at the beginning of the title (TITLE-FRONT), at the end of the title (TITLE-END), at the beginning of the snippet (SNIPPET-FRONT), at the end of the snippet (SNIPPET-END), and in URL. By studying the sample

news sites, we have the following observation:

Observation 2. If the publication time appears in the title or snippet of a SRR, it always appears at the very beginning or the end of the title or the snippet. But if it appears in URL, there is no such regularity.

The position information associated with each extracted Time Set is used to determine the location. Take a Time Set extracted from title as an example, if it is closer to the beginning of the title than to the end, it belongs to TITLE-FRONT; otherwise it belongs to TITLE-END. For the Time Sets in each of the TITLE-FRONT or SNIPPET-FRONT locations, we only keep the very first one (the one closest to the beginning). Similarly, for the Time Sets in each of the TITLE-END or SNIPPET-END groups, only the very last extracted time (the one closest to the end) will be kept. For the Time Sets extracted from the URL, they are simply merged into one Time Set. Actually, because of the special format of the URL, it is very rare that a URL contains multiple time information. In this way, for each result record at each location, there is only one Time Set left.

Next, the *frequency* of each location is counted (line 10 in Figure 5). The frequency of a location is the number of records that have the times extracted from the location. The location with the highest frequency is selected (line 11) as the potential place for holding the publication time (PTLoc). The rationale behind this selection policy is based on the following observation:

Observation 3. A news site usually displays the publication time at the same place in all of its result records.

Among the news sites collected in our testbed, 96% of them display the publication times for all the result records at the fixed location while only 4% display at different locations. If two or more locations have the same frequency, we break the tie by giving preference to title locations. This is because the title is usually very short and time information in it is most likely to be the publication time. The URL location has the lowest priority because the time in URL in many cases refers to the creation time of the archive that the news article belongs to. If the frequency of the selected location is above a threshold, we will set this location as the publication time location (lines 12-13). Currently, the frequency threshold is set at the half of the total number of records used to perform the extraction.

Once the publication time location is determined, for each result record, the real publication time will be selected from the extracted Time Set at this location. Remember that if the Time Set is a Disputed Time Set, there are multiple times in it. By looking at the Time Set individually and independently, it is difficult to determine which one is correct. To tackle this problem, we take the Time Sets at the publication time location of all result records into consideration. For each Undisputed Time Set, the only time in it is set to be the publication time for the corresponding result record, and the pattern is saved aside as an undisputed pattern (lines 14-18). Then, for each result record with a Disputed Time Set at the publication time location, the time with the pattern matching the publication time pattern will be set as the publication time for this record (lines 20-21). This solution to the equivocal interpretation problem is based on the following observation:

Observation 4. The publication times in all search result records from the same news site usually have the same pattern.

In other words, if the publication time for one result record can be determined, the times in other records that have the same pattern with this time and at the same location are highly likely to be the publication times for those records. Among our collected sample news sites that do display publication time in SRR, around 96% of them have at most two undisputed patterns. These patterns are called the publication time patterns for this news site.

Finally, for each of the remaining result records whose publication time has not been determined, we select one from all the Time Sets extracted in this SRR (lines 22-23). The geographical information is used to resolve the equivocal interpretations within a Time Set. If the site is in the region where there are some known conventions on displaying the time information, we filter out those times that do not follow the conventions. For example, if the site is in UK and we know that British people are used to put the day before the month, the extracted time with the pattern “MMddyyyy” can be eliminated. The geographical information usually can be obtained from the URL or IP address of the news site. For example, the domain name of UK sites usually end with “.uk”. If the ambiguity still exists, the most recent time will be selected as the publication time for this record. Then, the publication time is selected based on the priority of the time location. More specifically, for each result record, if there are times extracted in title, the most recent one among them will be selected as the publication time for this record. The times in snippet will be checked only if there is no time extracted from title. The last place to check is the URL.

5. Wrapper implementation

If a news search engine only returns one or very few results for a certain query, it will not be easy to figure out

the publication time for each result record even for the human beings. This is because even though the site uses a certain pattern to display the publication time information, it is difficult to recognize the pattern with a small number of SRRs. Let's consider another situation where the query is about a fresh news event. In this case, all the news stories on the result page returned by a news search engine may be published on the same day. If the expression of this day by chance can lead to several different interpretations (such as "05/04/03"), it would be very difficult to determine the correct date. However, if we knew that the news site uses a certain pattern to display its publication time and we knew what the pattern looks like in advance, then there would be no problem to handle the above situations.

In many applications in the news search engine domain, the employed search engines are usually known in advance. For example, the local component search engines are known to the news meta-search system. This provides us the chance to determine the publication time pattern for each used news search engine beforehand.

In order to determine the publication time pattern of a news search engine, we first collect some sample result pages from this search engine by submitting several queries to it. Then we merge the SRRs extracted from these result pages. The reasons for using multiple queries and to merge the SRRs are, first, we want the merged list to contain enough records to facilitate pattern analysis, and second, we want to increase the chance that all the time patterns that the site adopts are included. As a result, the chance of having disputed publication times can also be greatly reduced. The same methods for publication time extraction and identification as described in Section 3 and Section 4 are also used here except that these methods are applied to the merged list of SRRs. Once the potential publication times are extracted and the publication time location is determined, we can generalize the publication time patterns for this site. If multiple patterns are identified, they are organized based on their frequencies in descending order. The frequency of a pattern here is the number of records in the merged list having the publication time with this pattern.

We need a systematic way to describe whether the site has regular pattern or patterns for the publication time and if yes, where the publication time is located and what the pattern looks like. We call this description or expression as the *publication time wrapper* for this news search engine. In our work, the format of the wrapper is defined as following:

[pattern1] | [pattern2] | … @ [location].

The wrapper expression has two parts. The first part is the publication time patterns separated by the alternation operator "|" if multiple patterns exist. The order is based on their frequencies in the merged sample result record list. The second part is the publication time location. Two parts are connected by "@". We simply use "N/A" to denote that a site does not use any fixed pattern for publication time. Once the wrapper is built for a search engine, it is saved in a file together with other information about the search engine.

Given a set of result records returned from a news search engine, extracting the publication time with the support of the wrapper becomes straightforward and easy. The wrapper expression of this site is loaded from the saved wrapper file and the publication time location and the patterns are parsed out from the expression. Instead of checking all the places in the result record, we only consider the terms in the publication time location. Moreover, we only use the patterns listed in the wrapper to map the term sequence, one pattern at a time starting with the one with the highest frequency. If the location indicates a FRONT position, the first parsed time will be set as the publication time for this record and it is unnecessary to map the pattern with the rest of the terms. If the location is an END position, the last parsed time matching the pattern will be selected as the publication time. The patterns with lower frequencies in the list will be used only if no publication time can be extracted by using the patterns with higher frequencies. If no pattern matches the term sequence, we consider that there is no publication time for this result record.

If the wrapper indicates that the site has no fixed pattern for publication time (the expression is "N/A"), the same method described in Section 4 will be used directly to select the publication time from each single result record.

The advantages of the wrapper-supported extraction (***WSE***) are obvious. First, the extraction is independent of the number of result records on a particular result page. Second, resolving the ambiguity problem of time interpretation on the fly is no longer needed. We can directly pick the one matching the publication time pattern from the Disputed Time Set. Third, compared with the extraction without wrapper support method (***NonWSE***), fewer patterns need to be examined and they are used to map only a small portion of each result record, leading to reduced extraction time.

6. Experiments

6.1. Testbed

To test our methods, 50 news search engines are randomly selected. Some of them are US-based regional

newspapers such as *Houston Chronicle* and some are international news agents (e.g. *BBC news*). 10 queries are selected manually and they are listed in Table 2. Some queries are the general ones such as "President Bush". Some are about the hot events that happened during the time the experiments are conducted. Every query is submitted to all the search engines. For each search engine and each query, the first result page is collected and saved so that they will not be affected by any subsequent changes. Totally, we collected 411 result pages (some search engines return no result for some queries). The search result records on every result page are extracted using an automatic extraction tool called ViNTs [7] (note: ViNTs extracts SRRs only and it does not extract publication times). For each search engine, the 411 result pages are partitioned into two sets: **Dataset 1** contains those that are returned from odd numbered queries and **Dataset 2** contains the remaining result pages. Dataset 1 is used to build the publication time wrapper for the search engine while Dataset 2 is used to test the performance. The wrapper expressions are also saved locally. There are 205 result pages in Set 2 and the total number of records on these pages is 3,113. Among them, 3,080 records have publication time while 33 do not. The correctness of every extracted result is checked manually. An extraction is considered incorrect if (1) the result record has publication time but it is not extracted out, wrongly interpreted, or partially interpreted (e.g., only the date part is correct but the time in day is missing); or (2) the result record has no publication time but one time is assigned to it. The collected result records, the generated wrapper expressions, together with the correctness checking result, form our testbed.

Table 2. Testing queries

Q1	President Bush
Q2	hurricane Katrina
Q3	South Asian earthquake
Q4	Saddam Hussein trial
Q5	Harriet Miers
Q6	Iraq referendum
Q7	neo-nazi march violence
Q8	bird flu pandemic
Q9	high gas price
Q10	Chinese spacecraft

6.2. Evaluation criteria

We use *precision* and *recall* to measure the performance of our extraction methods. These two measures are widely used to evaluate the performance of information retrieval systems.

a) *Precision*: It is the ratio of the number of correctly extracted publication times over the total number of extracted publication times. For example, if 100 publication times are extracted but only 80 are correct, then the precision is 0.8.

b) *Recall*: It is the ratio of the number of correctly extracted publication times over the total number of correct publication times that need to be extracted. For example, if 100 publication times need to be extracted but only 90 are corrected extracted, then the recall in 0.9.

6.3. Result analysis

In the previous sections, we presented two publication time extraction techniques, one with wrapper support (denote ***WSE***) and one without wrapper support (denote ***NonWSE***). We first evaluate the precision and the recall of each method based on Dataset 2. The results are reported in Table 3, where the second column denotes the number of extracted publication times and the third column indicates the number of correctly extracted publication times. For the ***WSE*** method, publication times are extracted from 3,084 result records and 5 of them are incorrectly extracted, which corresponds to a precision of 99.84% (= 3079/3084) and a recall of 99.97% (= 3079/3080, where 3,080 is the number of SRRs that have publication times in Dataset 2). For the ***NonWSE*** method, the same number of publication times is extracted but there are 8 false extractions, which corresponds to a precision of 99.74% and a recall of 99.87%.

Table 3. Extraction performance comparison

	N	*n*	Precision	Recall
WSE	3084	3079	99.84%	99.97%
NonWSE	3084	3076	99.74%	99.87%

From Table 3, we can see that both methods are very effective. It seems that ***WSE*** is not significantly better than ***NonWSE*** in terms of the accuracy. The reason is that almost all the queries we selected for testing are too popular. We have 205 result pages and 3,113 result records in total for Dataset 2, which corresponds to an average of 15.2 records per result page. This means, for a given testing query, it is very likely that a search engine returns enough number of result records so that ***NonWSE*** is able to determine the publication location and the publication time patterns using these records dynamically. However, we made an interesting observation when we examined the result page returned from *Pittsburgh Post-Gazette* for query "bird flu pandemic". There are only 3 result records on this page. For every record,

NonWSE extracts one time with pattern "MMM dd yy" at SNIPPET-FRONT and another time with pattern "MM dd yy hh:mm:ss a" from URL. Since the frequencies of the two locations are the same (both are 3) and the SNIPPET has higher priority than URL, ***NonWSE*** treats the time at SNIPPET-FRONT as the publication time for each result record. But the real publication times should be the ones in URL. ***WSE***, on the other hand, with the support of the wrapper of this site (generalized by using the training queries that produced Dataset 1), correctly extracts the publication times from all of the three result records. In the real world, it is highly possible that a search engine may only return few result records for many queries, depending on its document collection, its document selection algorithm as well as the user query. The chance that ***NonWSE*** would make the same mistake as in the above example would be largely increased. From this example, it can be seen that the wrapper-supported method can indeed improve the extraction accuracy.

For each of the two methods, ***WSE*** and ***NonWSE***, we computed the percentage of testing sites for which 100% precision and 100% recall are achieved. The result is shown in Table 4. For ***WSE***, there are 48 sites achieved 100% precision and 49 sites achieved 100% recall. For ***NonWSE***, the numbers are 47 and 48 respectively. This experiment also shows that with the support of the pre-generated publication time wrapper, the extraction effectiveness can be improved.

Table 4. Search engine based comparison

	100% Precision	100% Recall
WSE	96%	98%
NonWSE	94%	96%

We also compared the two methods (***WSE*** and ***NonWSE***) in terms of their efficiency. Specifically, we compared their times needed to extract the publication times from the result records on one result page. It turns out that on average, it takes about **95 milliseconds** for ***NonWSE*** to process one page while ***WSE*** only needs about **41 milliseconds**, less than half of the time needed by ***NonWSE***. This experiment shows that the ***WSE*** method can improve not only the effectiveness but also the efficiency of extraction.

By examining the false extractions that each method made, we found that except the 3 records from *Pittsburgh Post-Gazette* as introduced earlier, two methods made the same mistakes on the remaining 5 result records. These mistakes happened on two situations. First, the publication time expression has no year information and it is the only temporal information in the SRR. In this case, the current year is assigned. However, by following the link to the article page, we found that the actual publication time is in 2003. How to more accurately determine the year without checking the news article is an issue we'd like to study in the future. The second situation is that, a result record contains some temporal information but none of them is the publication time. Moreover, the site does not use any fixed pattern to display the publication time (so the wrapper expression for this site is "N/A"). In this case, our methods select the most recent time from the high priority place as the publication time for this record. How to improve the accuracy under this kind of condition is another issue we are interested in.

7. Conclusion

In this paper, we studied the problem of extracting the publication time from the news searching results. The proposed method has several distinct features. First, we considered all kinds of variations of each time component's presentation format, which is ignored by many previous researches. Second, we made several important observations that are useful for the extraction. For example, we found that news search engines tend to display the publication time in the same place in the result record with the same pattern. This observation directly leads to our solution of using the position information to determine the publication time. Third, we handled the ambiguous interpretation problem by checking the time patterns across the publication location. We also proposed other techniques such as using the news site's regional conventions to resolve the issue. A wrapper implementation was proposed in this paper. The experimental results over 50 news search engines show that while both methods are effective, the method with wrapper support can not only improve the extraction quality but also the extraction efficiency.

In this paper, we considered extracting publication time from the search results returned from one source. An interesting and more challenging problem is to extract the publication time from search results originated from multiple sources. In this case, the formats of the times would be more diverse. We will look into this problem in the future.

A news search system, for many reasons, likes to change their formats to display the search results frequently. As an important fact embedded in the result record, the publication time location or pattern may also be changed subsequently. In this case, the previously built wrapper may not function any longer and it has to be rebuilt. However, how to determine the time when the wrapper should be rebuilt is an interesting issue. It is especially crucial in a large system where there are thousands of component new search engines. We also

plan to investigate this issue in the near future.

Acknowledgement: This work is supported in part by fundings from the following sources: Webscalers, NSF grants IIS-0414981, IIS-0414939 and CNS-0454298.

8. References

1. J. Allan, R. Gupta, and V. Khandelwal, "Temporal Summaries of News Topics", *Proc. of the ACM SIGIR conference on Research and development in information retrieval*, New Orleans, Louisiana, USA, 2001, pp.10-18.
2. Y. Rasolofo, D. Hawking, and J. Savoy, "Result Merging Strategies for a Current News Metasearcher", *Inf. Process. Manage*, 39(4), 2003, pp.581-609.
3. D. B. Koen and W. Bender, "Time Frames: Temporal Augmentation of the News", *IBM Systems Journal*, 39(3&4), 2000, pp. 597-616.
4. B. Sundheim and N. Chinchor, "Named Entity Task Definition, Version 2.0", *Proc. of the 6th Message Understanding Conference (MUC-6)*, 1995, pp.319-332.
5. N. Chinchor. "MUC-7 Information Extraction Task Definition, Version 5.1", *Proc. of the 7th Message Understanding Conference (MUC-7)*, 1998.
6. L. Ferro, I. Mani, B. Sundheim, and G. Wilson, "TIDES Temporal Annotation Guidelines", *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(1), 2004, pp.33-50.
7. H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu. "Fully Automatic Wrapper Generation for Search Engines", *Proc. of 14th World Wide Web Conference (WWW14)*, Chiba, Japan, May 2005, pp.66-75.