POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Metadata Extraction and Digital News Preservation

TESI DI LAUREA MAGISTRALE IN
COMPUTER SCIENCE AND ENGINEERING
INGEGNERIA INFORMATICA

Author: **Victor Henrique de Oliveira Santarosa Martins**

Student ID: 10785585

Advisor: Prof. Letizia Tanca

Co-advisor: Dr.Eng. Davide Piantella

Academic Year: 2022-23

Abstract

The Internet provides people with a huge amount of information that grows fast, but this rapid information production creates challenges for the access and preservation of digital news content. This information growth generates a situation where a lot of digital news became lost or unavailable.

Metadata extraction is a technique that can help to address these challenges by extracting relevant information from digital news articles, such as title, author, date, keywords, summary, images, etc.

Metadata extraction can also facilitate the generation of preservation metadata that can ensure the continued accessibility and usability of digital news content over time. Preservation metadata can include information about the content, format, rights, and technical characteristics of digital objects, and help to ensure their long-term management and interoperability.

The main objective of this research is to investigate the current state-of-the-art in metadata extraction and explore techniques that can be useful in the context of information preservation proposing a novel approach for metadata extraction from digital news articles, using a combination of the analyzed techniques.

The proposed approach involves the development of a system capable of extracting metadata from articles published on news websites and cataloguing them, ensuring their continued accessibility and long-term usability.

Keywords: Digital News, Metadata, Metadata Extraction, Information Preservation.

Abstract in italiano

Internet fornisce alle persone una grande quantità di informazioni che crescono sempre più velocemente, ma questa rapida produzione di informazioni crea sfide per l'accesso e la conservazione dei contenuti delle informazioni in formato digitale. Questa crescita genera una situazione di caos in cui molte notizie digitali vengono perse o diventano indisponibili.

L'estrazione dei metadati è una tecnica che può aiutare a risolvere queste sfide estraendo informazioni rilevanti dagli articoli in formato digitale, come titolo, autore, data, parole chiave, riassunto, immagini, ecc. L'estrazione dei metadati può anche facilitare la generazione di metadati di conservazione, che possono garantire l'accessibilità e l'utilizzabilità continue dei contenuti delle notizie digitali nel tempo. I metadati di conservazione possono includere informazioni sul contenuto, formato, diritti e caratteristiche tecniche degli oggetti digitali, e aiutano a garantire la loro gestione a lungo termine e l'interoperabilità.

L'obiettivo principale di questa ricerca è investigare lo stato attuale dell'arte nell'estrazione dei metadati ed esplorare tecniche che possono essere utili nel contesto della conservazione delle informazioni proponendo un nuovo approccio per l'estrazione dei metadati dagli articoli di notizie digitali, utilizzando una combinazione delle tecniche analizzate. L'approccio proposto prevede lo sviluppo di un sistema in grado di estrarre i metadati dagli articoli pubblicati sui siti di notizie e di catalogarli, garantendo la loro accessibilità continua e l'utilizzabilità a lungo termine.

Parole chiave: Notizie digitali, Metadati, Estrazione dei metadati, Conservazione delle informazioni.

Contents

1 Introduction

The Internet provides people with a huge amount of information, that grows faster every day; access to information has never been so easy and almost instantaneous.

On the other hand, this fast growth in information production is leading us to an information explosion problem, where, due to the information overload, users are unable to find the data meeting their needs in an efficient way.

When we think about digital news, at first sight, this explosion of information can appear to be positive since consumers will have greater access to information through the internet. But, first, this information explosion can lead to a decrease in quality and credibility of news since there is a pressure to produce more content and in a faster way. Also, there are problems related to the management of the information and preservation of news.

In 2018 some journalists from Columbia Journalism Review [1] made a research report about archiving practices and policies in digital news companies. They conducted interviews with individuals from 30 news organizations and found that most of the interviewed companies did not have any strategies for preserving their digital content. Many interviewees considered backup and storage in platforms like Google Drive as synonymous with archiving.

However, there is an important difference between backup and archiving; a backup is focused on saving, restore, and recover data, rather than ensure long-term access to it. Backing up information is not enough to ensure that information will be individually available and accessible, due to lack in organization or potential hazards generated by technological evolution.

This information growth generates, together with the lack of preservation strategies coming from the news organization, a situation where a lot of digital news became lost or virtually unavailable. The content will still be on the Internet, but the overload of available information makes impossible to recover it.

News preservation is not only a matter of historical memory, but also of information access.

1.1 Metadata Extraction and Digital News Preservation

Metadata is defined as data providing information about one or more aspects of the data; it is used to summarize basic information about data that can make tracking and working with specific data easier.

Metadata extraction is the process of automatically extracting relevant information from some content, e.g., in a digital news article we could extract information as the title, author, date, keywords, summary, images, etc.

This can help with the problem of ephemerality of news by enabling users to quickly access and compare different sources and perspectives on a topic, to filter and organize news content according to their preferences and needs, and to evaluate the credibility and quality of news content based on their metadata.

Digital newspapers preservation is the process of ensuring the continued accessibility and usability of digital newspapers over time. Metadata extraction can facilitate the preservation and archiving of digital news content for future reference and analysis by generating preservation metadata. It can include information about the content, format, copyrights, and technical characteristics of digital objects. Preservation metadata can help to ensure the long-term usability and management of digital objects, as well as their interoperability with other systems or standards.

1.2 Objectives

The focus of this thesis is on Metadata Extraction, specifically for digital newspapers. Our goal is to explore various approaches and techniques that enable us to extract meaningful metadata from digital news articles, thereby improving their digital preservation. We aim to develop a system capable of extracting metadata from articles published on news websites and cataloging them, ensuring their continued accessibility and long-term usability.

Our proposed approach involves a combination of HTML parsing and extraction, using HTML tags to extract useful information, as well as popular metadata extraction techniques such as Natural Language Processing, Regular Expressions, and Template-based matching. By developing this new approach, we hope to improve the process of metadata extraction and enhance the preservation of digital news articles.

1.3 Methodology

The approach employed in this thesis comprised the following components:

- A preliminary literature review to explore existing methods for extracting metadata from documents. This involved comprehending the various techniques, their strengths, and limitations.
- An evaluation of the previously mentioned aspects to determine the most suitable techniques for our purposes. This includes examining the possibility of combining different techniques to propose a novel approach.
- A deep analysis of HTML source code of some popular news websites to find similarities and differences that can be exploited to improve your metadata extraction techniques.
- The definition of a proposed system for extracting meaningful metadata from digital news published on journalistic websites, utilizing a combination of the previously presented approaches and techniques.

- A real implementation of the proposed system to validate their capacity to extract metadata from digital news articles.

1.4 Thesis Organization

This thesis has the following structure. Chapter 2 presents a state-of-the-art review, presenting some relevant notions to the development of the thesis. Chapter 3 presents the goals, define requirements of the thesis, and proposes a System Architecture. Chapter 4 shows a deep analysis of HTML documents proposing approaches to extract the desired information. Chapter 5 shows and details the implemented system and discusses their results. Chapter 6 gives a summary of all the work done and presents some future work plans.

2 State of the Art

In this section, we will define and explain key concepts to help readers better understand the work presented here. First, we will introduce the concept of metadata and discuss various techniques for metadata extraction, including their advantages and disadvantages. Next, we will explore how HTML features can be used to extract meaningful data from webpages and discuss web crawling as a method for obtaining news articles. Finally, we will address the importance of digital preservation in a world where vast amounts of data are generated every day.

2.1 Structured, Semi-structured and Unstructured Data

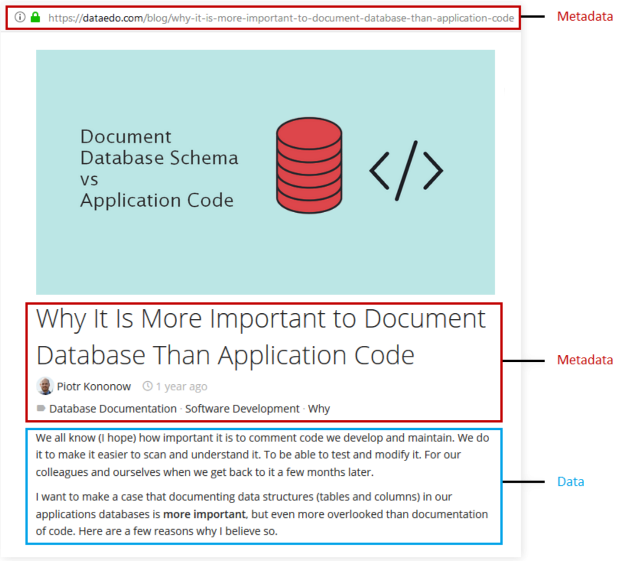
Structured data is data that has a fixed and defined format such as an SQL Table or an Excel Spreadsheet. It is easy to store, analyze and query, but due to its fixed form it cannot be able to capture all the complexities of the real world.

Semi-Structured data is data that has some elements (like tags) that give some structure and meaning to the information but without losing flexibility in the way data is organized and stored. HTML web pages are an example of semi-structured data.

Unstructured data is data that has no predefined structures such as a plain text or an image. Unstructured data is diverse and can be very valuable, but it may be difficult to store, process and analyze. A process of mapping unstructured data to structured data can be needed to manage these data effectively in an efficient manner.

2.2 Metadata

Metadata is data that provides information about other data, classically defined as ***data about data***. It is a description of the data that helps to organize, find, and give some context to it. Metadata can be found everywhere in many different forms and almost every content in a system comes with metadata. For example, metadata can include information about an item like creation date, author, title, and other details. [2]



https://dataedo.com/blog/why-it-is-more-important-to-document-database-than-application-code

Why It Is More Important to Document Database Than Application Code

Piotr Kononow 1 year ago

Database Documentation · Software Development · Why

We all know (I hope) how important it is to comment code we develop and maintain. We do it to make it easier to scan and understand it. To be able to test and modify it. For our colleagues and ourselves when we get back to it a few months later.

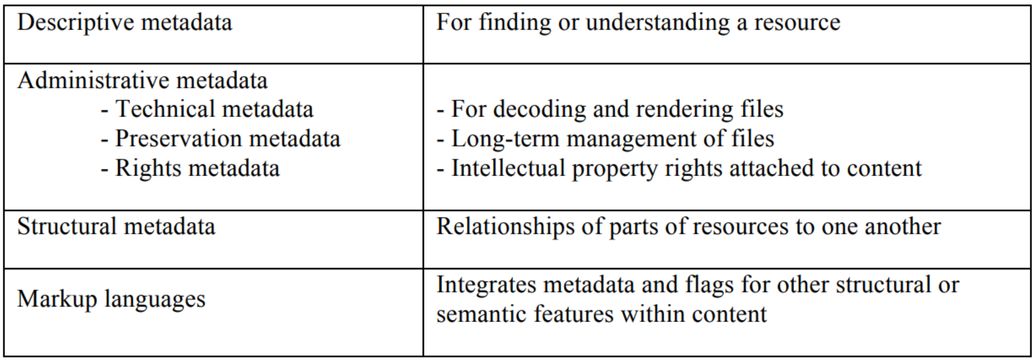
I want to make a case that documenting data structures (tables and columns) in our applications databases is **more important**, but even more overlooked than documentation of code. Here are a few reasons why I believe so.

Figure 1 - Example of Metadata in a Blog Post [3]

For metadata to be useful it should at least be structured to some degree. It is collected to fulfill some kind of purpose and is this notion of structure that turns raw information into actionable metadata. It is collected and stored in a way to model the most important features of the data it describes, and this structure is what makes metadata such a valuable tool for organizing and managing information.

A Metadata schema is the set of rules that ensure this structure to metadata, like defining that the information about the creation of an object should be written as YYYY-MM-DD. The schema defines the elements of data that should be provided – or extracted – and how it should be provided. At same time the elements in a metadata schema are the category of descriptions that can be made about a resource, like a title, creator, and date. [4]

2.2.1 Types of Metadata



Descriptive metadata	For finding or understanding a resource
Administrative metadata - Technical metadata - Preservation metadata - Rights metadata	- For decoding and rendering files - Long-term management of files - Intellectual property rights attached to content
Structural metadata	Relationships of parts of resources to one another
Markup languages	Integrates metadata and flags for other structural or semantic features within content

Figure 2 - Types of Metadata [2]

Descriptive Metadata are descriptive information about a resource. This kind of metadata includes elements like Title, Creator, Publication Date, Keywords. Their primary usage is for understanding, organizing, and finding a resource.

Administrative Metadata refers to information's needed to administrate and use a resource, or even information related to its creation. It can be separated into Technical Metadata, which are related to digital files and contains the elements about what is needed to decode and read these files (e.g., what is the file type or the version). Preservation Metadata are the elements needed to preserve information through time, like a checksum to verify the integrity of the information; Rights Metadata which details the intellectual property of an information and who can have access to it.

Structural Metadata describes relationships between objects/resources and can be helpful for navigation.

Finally, another type of metadata is *markup languages* (e.g., HTML). It mixes metadata and content all together. In HTML for example *tags* inserted in the content can describe it, denoting for example that some part of the content is a Title, a paragraph, or an image. It can also give style information about a content like its font size and color.

2.3 Metadata Extraction

Metadata Extraction is the process of automatically extracting metadata from different sources of data. We can rely on many different tools and techniques to do this automatic extraction from different kinds of data.

The manual extraction of metadata from files can be very time consuming in a data-driver world like ours; thousands of new information are produced every day and it is basically impossible to manually analyze all of them to extract meaningful metadata. In that sense the usage of tools to automatic extract these metadata become very relevant in many areas.

Several methods for automatic extraction have been proposed; among them *Machine Learning Systems* and *Rule-based Systems* are the most popular ones.

Regular Expressions and *Rule-based techniques* can achieve a good performance, don't require any training and are easier to implement; but can be less adaptative, depending more on the domain in which it is used. Also, the need for an expert to set the rules (which can be very complex) can limit the usage of these techniques. [5]

On the other side, *Machine Learning techniques* can be more adaptable and can lead to promising results but can be harder to implement and require some labeled data to do the training. Also, besides the fact that it could be used in theory in any kind of document, machine learning algorithms tend to have a decline in efficiency when the heterogeneity of the collection increases.

For a very heterogenous collection of data both techniques will have their limitations and will not perform very well; revealing the necessity of bringing multiples techniques together to achieve a better performance during extractions.

In the next sub-sections, we will bring details of each of these techniques and present existent research about them.

2.3.1 Machine Learning Techniques

Machine Learning (ML) methods offer and robust and adaptable way to do the automatic extraction of metadata.

We can find lots of research using the most varied ML techniques (SVM, Deep Learning and others) to do automatic metadata extraction, but in a more general way the ML techniques that we can use are:

- **General Classification:** can be used to extract and determine the roles played by a fragment in a document. In summary, assign a category to each fragment in a text.
- **Sequence Classification:** for the classification we are interested in analyze a sequence of text fragments instead of an independent fragment.
- **Clustering:** group a set of text fragments (and other set of objects) into clusters. The fragments in the same cluster have similar characteristics.

Han et al. [5] proposed a Support Vector Machine (SVM) classification-based method to extract metadata from header part of research papers. The proposed method first classifies each line of the analyzed text into one or more pre-defined classes (general classification). Then, an interactive method is used to improve the classification by using the classification of its neighbors (sequence classification). Finally, metadata extraction is done by seeking the best classification of each line. In summary, extracting metadata can be considered a "classification problem". To extract features for classification, both word and line-specific features are used for classifying text lines into one or more classes. A rule-based and context-dependent word clustering method was designed to group similar words in clusters and use these clusters as features for the ML classification.

Liu et al. [6] introduced deep learning into the task of extract metadata from documents. They utilize not only text content but also image content and vision information (e.g., text position, font, and layout) to improve the quality of the metadata extracted. These deep learning-based approaches can automatically learn the feature representation of metadata during training, which reduces the manual work of extracting these features.

2.3.2 Regular Expressions and Rule-based Techniques

Rule-based methods try to do the automatic extraction of metadata using several rules to identify the layout of a text and extract the metadata from it. Keywords, spatial and visual information like the font size and the position of a text in a document can be used as a guide to the metadata extraction process.

It’s biggest advantage over Machine Learning's techniques is that you don't need to generate labeled training data, which is very time-consuming and costly. Although, it is important to note that since rule-based techniques uses the document layout to do the extraction, it is not very adaptable to a heterogeneous collection of data. [7]

Zhixin et al. [7] proposed a rule-based framework for automatic extraction of metadata (e.g., titles, authors, and abstracts) from scientific papers. After collecting articles from the internet, they convert it to Text files or XML files (the conversion to XML can be more interesting because it has some format information about the text that can help the extraction to become more accurate). Then they defined some rules to identify if some file is a scientific article or not (mostly using rules to find some keywords). Finally, they extract the metadata using their proposed algorithm they rely at most in spatial information; they defined a set of rules that explore spatial properties on the layout of scientific papers. One point of attention here is that there are always some exceptions in the layout structure and to cover these exceptions the rules can become very complex.

Ojokoh et al. [8] proposed a model for metadata extraction from general documents that combines segmentations by keywords and patterns matching techniques. The main idea is to extract the structure of the document using *keywords* together with the use of regular expressions to extract some set of metadata.

A layout segmentation captures the divisions of the document's logical structure. It can be done in three ways:

- **Segmentation by spacing:** a document can be separated using spacing information into several areas which can be text, images, or tables for example.
- **Segmentation by style difference:** separate text areas where the styles are clearly different from those on other areas (e.g., different font size, bold text).
- **Segmentation by keyword:** usage of special keywords (e.g., abstract, references, summary) to segment certain areas of the document.

Their proposed system is based on first segmenting the document using keywords and then locating the keywords that are associated with some kind of metadata and locating the document parts corresponding to this keyword. This approach can be challenging for parts like titles because these parts usually are not labelled, so we cannot use segmentations by keywords; but it is possible to think in segmentation by style difference or pattern matching for example.

Many of the extraction techniques presented focus on extract data from PDFs and other similar formats of data, in contrast, *Tang et al.* [9] proposed a system that focus on web content by utilizing structural and semi-structural features of HTML in association with regular expressions to extract metadata in an effective and accurate way. First, they clean the web page removing irrelevant HTML code; then they analyze the remaining HTML trying to find and extract specific segments of code; finally, they extract the desired metadata using regular expressions.

Giuffrida et al. [10] also developed a metadata extraction system based on spatial/visual knowledge. First, they translate the input document to a set of strings of text annotated with spatial and visual information (like the position, page number, and font size); then a set of rules are used to extract metadata from those set of strings. One important thing to notice is that there is an implicit *fuzziness* involved in metadata extraction based on visual information (e.g., not always the titles is using the largest font in the document); to deal with this type of exception more complex rules are needed.

2.3.3 Natural Language Processing (NLP)

Natural Language Processing (NLP) refers to an area in Computer Science focused in giving computers the ability to understand text and spoken words in a similar way human beings can. Basically, NLP reveals the structure and meaning of text and allows us to analyze texts and extract information from them.

NLP works by finding relationships between parts of language – for example, the letters, words, and sentences of a text. Some of the tasks that can be solved with NLP are Sentiment Analysis, Text Translation, Named entity recognition, Summarization, and many others.

We can use NLP to extract metadata from documents in a more domain-independent way. In many cases, when we are talking about rule-based systems, we focus on rules that are in general very specific for certain domains of data.

Yilmazel et al. [11] proposed a system, called *MetaExtract,* to automatically assign metadata using Natural Language Processing extraction techniques. Their system compiles the output of three distinct modules: eQuery, HTML-based Extraction and a Keyword Generator Module. The *eQuery* module used NLP to extract terms and phrases found within single sentences; it is a rule-based system that uses parsing rules and multiple levels of NLP tagging. The *HTML-based Extraction* module uses the HTML structure to determine where the contents of the element can be and then it compares the text in that location to a list of clue words to determine which metadata element is present. The *Keyword Generator* module uses the previous HTML-based module to identify which section of the document should be used to process keywords.

Paik et al. [12] described a metadata extraction technique based on NLP which extracts personalized information from email communications. Their system, called <*!metaMarker*>, uses NLP and Machine Learning to process textual data and extract both explicit and implicit metadata. Utilizing both domain-independent and domain-dependent NLP techniques they were able to represent each sentence of the input email as a feature vector and then extract explicit metadata.

2.3.4 Template-Based Techniques

So far, we have presented two main approaches to extract metadata, Machine Learning Systems and Rules-based Systems. Both can be very powerful and efficient in extracting metadata from multiples data domains but mostly when these domains are composed of documents with a certain level of homogeneity. For a very heterogeneous collection of data both techniques will have their limitations and will not perform very well.

Flynn et al. [13] proposed a system that still relies in rule-based techniques to extract metadata but using a two-part system that first classifies the documents into groups of similar layouts. Then, associate with each group a template of rules that should be used to extract the metadata from each kind of layout. This approach allowed them to achieve good performance in the metadata extraction even in heterogeneous collections without the need of creating super complex sets of rules.

2.4 HTML Parsing and Extraction

HTML (HyperText Markup Language) is a markup language that defines the meaning and structure of web content. It uses "markup tags" to annotate the content (e.g., text, images, links) for display in a Web Browser. Basically, HTML defines the structure of a web page. [14]

HTML can be considered as semi-structured data, since it has tags and elements that give the data some structure, but it also has some variability in how the data is organized and which tags are used.

HTML parsing is the process of read and interpret a HTML document and generate a modified version of the documents, like a *Document Object Model (DOM) tree.*

HTML Extraction refers to the process of extracting data, such as texts, images, or other elements, from HTML documents. Usually, HTML parsing is a prerequisite for HTML Extraction. The extraction can also be done using other methods like regular expressions.

A DOM Tree is a tree structure where the nodes represent a HTML document's content. For every HTML document it is possible to extract a DOM tree representation. The DOM tree allows us to parse the HTML document get a better representation of each of their elements.

Basically, the DOM defines the logical structure of a documents in the format of a tree. When we think about information extraction this can be very useful to facilitate the extraction of certain elements of a document.

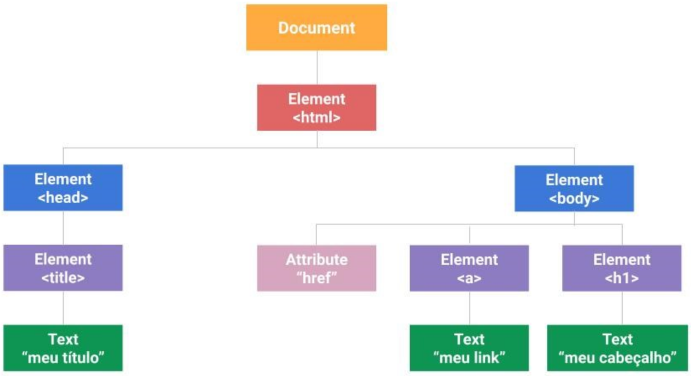


Figure 3 - DOM tree representation of a HTML Document

3 Project Description

3.1 Concepts

When we think about journalistic websites, despite some layout differences, most of them follow a similar structure of organization and presentation of content to the reader. They tend to have a similar layout, with the title at the top in a larger font size, followed by a subtitle.

Considering this, the usage of regular expression and rule-based techniques can be employed to extract metadata by exploiting these layout similarities. We believe that relatively simple rules could extract relevant information from a news article.

HTML parsing and extraction can be used to extract raw text from a web page, with only some style and layout information annotated to it. This can be used as input for rule-based extraction.

In addition, we can use HTML parsing and extraction approaches to directly extract relevant metadata such as titles and authors from news content. The metadata extracted here can be similar to what we are able to extract using regular expressions, but we believe that this redundancy is a way to improve the accuracy of the proposed system.

Although there may be some similarities between how journalists present their news on websites, there are thousands of news producers in the world, and it is inevitable that differences will appear. This can lead to a great increase in the complexity of the rules and regular expressions needed to extract the desired metadata. In that sense, it is worth considering a template-based approach that focuses on encapsulating rule-based extraction by defining specificities of each extraction source and allowing the rules to be simpler.

Finally, a relevant piece of metadata to extract when considering news preservation is keywords. Keywords can help categorize and better organize a digital news article archive. For this type of extraction, we believe that the best approach would be to use Natural Language Processing (NLP) techniques to extract these keywords from the main text of the news article. By using NLP techniques for keyword extraction, it is possible to efficiently organize and categorize large collections of news articles. With NLP we can automatically identify and extract the most relevant words and phrases from a text, providing a summary of its content.

3.2 Goals and Requirements

The main goal of this thesis is to create a system that can help in the preservation of digital news by means of the extraction of meaningful metadata. This proposed system could be used by news companies or individuals to archive news in a way that ensures long-term usability and access.

The objective is to have a system that can receive a URL of a digital news and is able to extract the following metadata:

- Title
- Subtitle
- Author
- Publication Date
- Keywords
- Website name
- URL
- Copyright

The idea is that it will use a combination of multiple technologies to improve the quality of the extract metadata, being able to extract the desired metadata that was previously presented.

We aim to explore the semi-structural nature of HTML, understanding which Tags or Class names are commonly associated with some metadata. This will allow us to extract various of the metadata that we desire using a generic extractor that can fit in multiple journalist websites.

To improve beyond this, we want to add regular expressions techniques to extract metadata that we were not able to extract directly from the HTML, for example parse a publication date that was wrote as text in nonstandard format.

Also, to focus on keeping a good extraction performance even between different sources we will focus on a template-based system, where we will be able to define in templates some specificities of some webpage (or a group of webpages).

The idea is that the template will not directly change the way that the system will work but instead it will allow the user to set some custom configurations in the system without the need to change the source code. With a template we could define setting related to language and region of a website. We could set the class names that the parser should look up for to find the author (in English would be something like *'author'* or '*byline'* but in italian could be something like *'autore')* or set how the dates should be parsed (e.g., considering day first or month first).

To generate keywords related to the news article, since not every journalistic web site puts keywords in their article, we propose to use NLP techniques. By using Natural Language Processing, we aim to process the textual content of a news article and from that extract keyword related to it.

```
{
  "author": {
    "classNames": ["author", "byline", "from"],
    "delimiters": [" and ", ", ", " & "],
    "prefixes": ["by", "edited by"]
  },
  "copyright": {
    "classNames": ["copyright"],
    "prefix": ["© "],
    "suffixes": [
      "all rights reserved",
      "copyright"
    ]
  },
  "datetime": {
    "classNames": ["timestamp", "date", "time", "publish_date"],
```

Figure 4 - Template example

3.3 System Architecture

The proposed system will be composed by a set of key components, which are:

HTTP Request Module: This module is responsible for downloading webpages. It receives the URL and downloads the web page and saves it locally. It will also append to every webpage a <meta> tag containing the URL of the page.

Template selector module: This module will be responsible for executing the extraction using the correct template.

HTML Parsing Modules: This module is responsible for reading and interpreting HTML. We will have a specific parser that will clean some useless things from the source html like <style> and <script> tags and extract the snippets of content that can contain any data. Also, we will have a base parser to extract the text of the news.

Metadata Extraction Module: This module will receive the parsed html data from the previous module and extract the available metadata by using multiple approaches. It will use the html tags and their class names, the position of an element in the pages and regular expressions.

Metadata combination module: This module will work as a helper to *Metadata Extraction Module*, applying rules to the metadata candidates that were found and choosing the best ones.

NLP Keyword Generator Module: By using the extracted news content this module will generate the keywords for the news article.

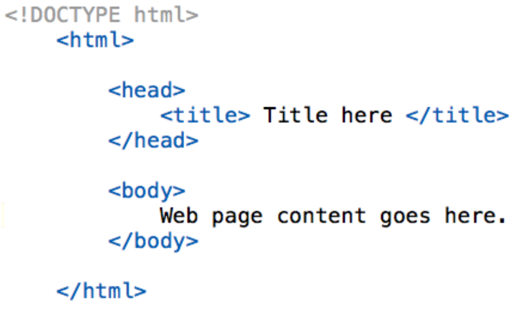
4 Data Analysis and Extraction

In this section we will focus on analyzing and exploring the HTML source code of digital news articles from some journalistic websites.

We aim to find similarities and differences between them, trying to understand which html tags or class names are generally associated with each kind of metadata.

In a first look we can find that the websites have a similar structure to present a news article.

A basic structure of HTML documents is composed of two parts: <head> and <body>.



```
<!DOCTYPE html>
    <html>

        <head>
            <title> Title here </title>
        </head>

        <body>
            Web page content goes here.
        </body>

    </html>
```

Figure 5 - HTML Basic Structure

The <head> part can provide general information (and metadata) about the document, including their title, description, keywords, and some metadata related to social networks.

In the head it is generally possible to find the title by looking for the <title> tag, but since this is the title presented on the title bar of the browser it can be slightly different from the news article document, in some of the analyzed page this title was a composition of the news title and the website name (*"{newsTitle} – {webSiteName}"*).

By using the <meta>, we can find some of the desired metadata like title, description, and webpage URL. The metadata that we were able to extract from these <meta> were not the same between each source, in all of them we could extract the description but only some of them had tags related to keywords, author or the website URL.

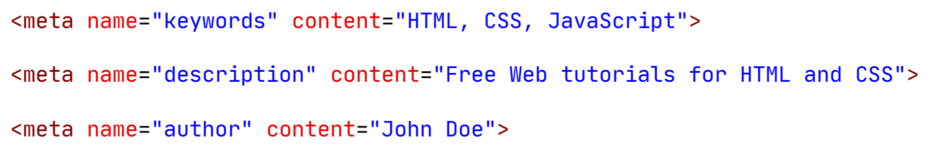
The <body> part surrounds the actual content of the page; it is this part that will be displayed in the browser. This part varies between each web site and can be composed of various parts. Two important parts from where we can extract the metadata that we are looking for are the ones surrounded by the tags <main> and <footer>.

The tag <main> surrounds the actual news article content. By parsing and cleaning the useless html tags from this part we can effectively extract the textual content of the news article. From here we can not only extract the title, subtitle, publication date and author but also, we can extract the actual news text.

4.1 The <head> section

4.1.1 The tag <meta>

The <meta> element in HTML represents a metadata about the document and will not be displayed on the page. <meta> tags always go inside the <head> element and are usually used to specify information like page description, keywords, author and etc.



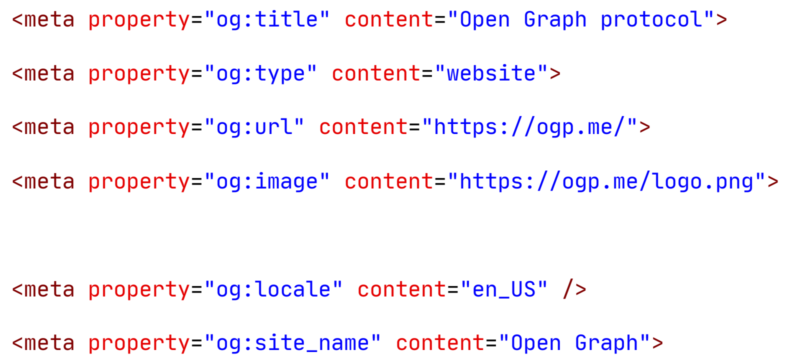
```
<meta name="keywords" content="HTML, CSS, JavaScript">

<meta name="description" content="Free Web tutorials for HTML and CSS">

<meta name="author" content="John Doe">
```

Figure 6 - Meta tags usage examples

Using this <meta> we can extract some of the desired metadata from the news article. The tag related to description were more common between the analyzed websites but some of them also presented tags related to keywords or authors.

4.1.2 Open Graph tags

Open Graph is an internet protocol to standardize the use of metadata within a webpage to represent their content. The open graph uses <meta> tags with some specific *property* value to represent some basic and optional metadata.



```
<meta property="og:title" content="Open Graph protocol">
<meta property="og:type" content="website">
<meta property="og:url" content="https://ogp.me/">
<meta property="og:image" content="https://ogp.me/logo.png">

<meta property="og:locale" content="en_US" />
<meta property="og:site_name" content="Open Graph">
```

Figure 7 - Open Graph tags examples

There is four basic and mandatory metadata in open graph:

- **Title** - The title of your object as it should appear within the graph.
- **Type** - The type of object.
- **Image** - An image URL which should represent your object within the graph.
- **URL** - The canonical URL of your object.

And some optional metadata that can be interesting for us like:

- **Description** – The description of object.
- **Site Name** – The overall website name.
- **Locale** – The locale where this page is available in and their language.

In the journalistic websites that implement Open Graph tags we can easily use them to extract some of the desired metadata like title, descriptions (or subtitle in the case of a news article), URL and sometimes even the website name.

4.2 Extraction approaches

Besides all the possible extraction using the information available in the `<head>` section of a HTML page, this is not always enough to extract all the metadata that we desire. Considering that, we aim to present some alternative approaches to extract the metadata from the HTML page of a news article using the <body> section of the page, which is where the actual content that is shown in the page is located.

4.2.1 Title (or Headline)

The Title can be extracted from the `<body>` part of the page, being surrounded by a `<h1>` tag. Some sites use a `<h1>` tag for other information that is not the title, so this approach will give us a list of metadata candidates that we will need to analyze.

It can also be found in the `<header>` part, surrounded by a `<title>` tag, but this title is generally a composition of the news title and the website name. Since that `<title>` tag contains the information that will be showed in the top of the browser some websites put on this tag the web site name instead of the news title, although it was rare between the analyzed sites.

Sometimes, it can also be found in a `<meta>` tag identified by an attribute `{"name": "title”}` or in an open graph tag identified by and attribute `{"property": "og:title”}`.

Considering that, the probably best approach to find the correct title metadata will be use a combination between the extracted candidates, for example trying to match one of the metadata candidates from the `<h1>` with the ones extracted looking on the other tags.

4.2.2 Description (or Subtitle)

The first point to notice was that not all the examined news articles had a subtitle. Although, some of them besides missing the subtitle in the news body had a description `<meta>` tag in the head that we could use.

When a subtitle exists, it is sometimes surrounded by <h2> tag but in some cases, it was not identified by any special tag, being represented as a normal text. Anyway, in all cases the subtitle was found right above the title, so we could use consider their position to find and extract it.

Finally, it can be found in a <meta> tag in head identified as *description*.

4.2.3 Authors

The authors of the article can be generally found right above the headline or subheadline in a <div> block together with the publication date. Besides that, because of the fact that it is generally together with the publication date, can be a hard task to obtain it using their position in the page.

Also, there are notable differences in how the authors are presented on each web site. In some there is the presence of some prefixes like *the* word *'by'* and the authors are separated by delimiters like a comma, but in others don't.

Considering that a good approach could be search by tags that have a specific class name (e.g., 'author', 'name', 'creator', 'byline').

In some articles there were <meta> tags about the authors.

4.2.4 Publish Date

As said before the publish date is generally in a <div> block together with the authors. In some websites it can be identified by the tag <time>, but not in all of them; in some of them it is inside a <div> or <span> with a class name referencing word like '*publish date*', '*timestamp*' or '*time*'.

Also, some of the analyzed pages presented <meta> and open graph tags about the publish time of the article.

Finally, a last approach is trying to extract the data from the URL by using regular expressions.

For all the previous approaches the datetime extraction can be very problematic due to the many differences in the way that the date is presented on each web page. Also, there are also differences caused by regional differences, like use month before the day, and the time zone.

4.2.5 Keywords

In some of the analyzed websites we were able to extract keywords from a `<meta>` in the head part, but in some there was not any data about keywords.

For the articles where the keywords were not found an approach is to use NLP techniques to generate the keywords from the extracted news. For that we need to extract the textual content of the article.

4.2.6 URL

As we stated before, during the article download we inserted a `<meta>` tag with the URL, so we can use it here to extract that information. Also, in the original HTML source we can generally find the URL in the `<head>` part, surrounded by a `<link>` tag.

4.2.7 Copyright

The webpage copyright information can be generally found in the footer of the page inside the body but the correct positions or the way that is presented varies a lot. So, the best approach would be search in the tags by the one with a class name referencing *'copyright'* or even search directly in the text by words like *'all rights reserved'* and *'copyright.'*

5 Implementation

In this section we present the implementation of a proof of concept that aims to demonstrate that it is possible to extract the desired metadata using the approaches presented in the previous section.

Among all the available open-source technologies available we have chosen to do our implementations in *Python* and mostly using an open-source library called *Beautiful Soup* [15] together with the python's built-in html parser [16].

Beautiful Soup is a Python library for pulling data out of HTML by using some compatible parsers and provides ways to navigate and search through the html source code. It was chosen due to its popularity, good documentations and powerful tools that allowed us to scrap the html source in different ways.

The implemented systems were made as follows:

First, we created a **HTTP Request Module** responsible for downloading a html page given a URL and save it in local storage. Given a URL, the module checks if the file already exists in the local storage and, if not. downloads it.

Then the **HTML Parsing Module** parses the downloaded html and cleans it, removing unnecessary tags, mostly tags related to scripts, styles, and links. We believe that a deeper analysis of the HTML source code could lead us to find ways to improve the html cleaning part and find more tags that are useless for our purposes.

Given the parsed HTML, for each one of the desired metadata we developed a **Metadata Extraction Module** implementing the approaches defined in section 4.2. The extractor outputs for each of the used approaches a list of the metadata candidates.

Finally, a **Metadata Combination Module** gets the metadata candidates generated for each desired metadata and, by applying some predefined rules, choose the best one to assign as the correct metadata.

This implemented workflow allows us to, given a URL, download an article and generate the metadata for it.

The source code of the developed implementation can be found on the following GitHub public repository: newsMetadataExtractor

5.1 Implementation Results

The proposed implementation led us to promising results. We were able to extract almost all the desired metadata for a different number of websites.

Our test was made by using thirteen different news websites from four different countries. Seven of then written in English, six from United States and one from United Kingdom; Five of them from Brazil and written in Portuguese; and one of them from Italy and written in italian.

The performance of the system was satisfactory, being able to extract Title, Headline, URL and Web Site name from all the Articles. Considering the authors, in only one of them we were not able to extract it.

For the copyright we also presented a good performance, missing only some of them but in some of the articles the values extracted presented some fuzziness.

In the Publish Date field we had bigger problems and mistakes. In articles from United States, we have parsing problems with the parsing mixing Day and Month in ambiguous dates. In other case we were able to find a publish date value but were not able to parse it due to the usage of a nonstandard formatting.

Nonetheless, after carrying out a deeper analysis of the errors, we tend to believe that most of them would be solved by the implementation of the proposed template module. Almost all errors related to the extraction of the publish date were related to regional differences in represent a date and with our proposed template this error would not happen.

Also, some improvement in our parsing module, cleaning more useless tags from the HTML would reduce the fuzziness found in some extractions.

Talking about the keywords, most websites do not have a <meta> about it, so only a few articles had this metadata extracted. But the implementation of the proposed NLP Keyword Generator Module would solve that.

6 Conclusion

6.1 Summary

In this thesis, we have presented an approach for metadata extraction from digital news articles using a combination of techniques (HTML parsing and extraction, natural language processing, regular expressions, and template-based matching).

We proposed and partially developed a system that can extract and catalog metadata from digital news websites, improving their digital preservation and accessibility. The system was evaluated using multiples digital news sources from different countries and languages and achieved good results for most of the desired metadata fields.

We also have discussed the challenges and limitations of our approach, such as parsing ambiguous or nonstandard dates, and handling fuzziness in the extracted information.

We hope that our work can contribute to the advancement of metadata extraction techniques and the preservation of digital news articles.

6.2 Future Work

As future work we aim to finish our implementation, developing the template module and the NLP keyword generator module, and evaluating their impact on the metadata extraction performance and accuracy.

Also, we aim to develop a complete archiving system that will use our extraction implementation to archive the news articles. Developing a user interface to simplify the usage.

Bibliography

[1] S. Ringel and A. Woodall, "A Public Record at Risk: The Dire State of News Archiving in the Digital Age," Columbia Journalism Review, 28 March 2019. [Online]. Available: https://www.cjr.org/tow_center_reports/the-dire-state-of-news-archiving-in-the-digital-age.php. [Accessed 24 July 2023].

[2] J. Riley, Understanding Metadata: What is Metadata, and What is it For?: A Primer, Baltimore: NISO, 2017.

[3] P. Kononow, "Dataedo," 16 September 2023. [Online]. Available: https://dataedo.com/kb/data-glossary/what-is-metadata. [Accessed 18 July 2023].

[4] J. Pomerantz, Metadata, MIT Press, 2015.

[5] H. Han, E. Manavoglu, C. L. Giles, H. Zha, Z. Zhang and E. A. Fox, "Automatic Document Metadata Extraction using Support Vector Machines," in *2003 Joint Conference on Digital Libraries, 2003. Proceedings*, Houston, TX, USA, 2003.

[6] R. Liu, L. Gao, D. An, Z. Jiang and Z. Tang, "Automatic Document Metadata Extraction Based on Deep Networks," in *NLPCC 2017: Natural Language Processing and Chinese Computing*, Beijing, China, Springer, Cham, 2017, p. 305–317.

[7] Z. Guo and H. Jin, "A Rule-Based Framework of Metadata Extraction from Scientific Papers," in *2011 10th International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, Wuxi, China, 2011.

[8] B. Adefowoke Ojokoh, O. Sunday Adewale and S. Oluwole Falaki, "Automated document metadata extractionAutomated document metadata extraction," *Journal of Information Science,* vol. 35, no. 5, pp. 563-570, 2009.

[9] X. Tang, Q. Zeng, Q. Cui and W. Zeze, "Regular expression-based reference metadata extraction from the web," in *2010 IEEE 2nd Symposium on Web Society*, Beijing, China, 2010.

[10] G. Giuffrida, E. C. Shek and J. Yang, "Knowledge-based metadata extraction from PostScript files," in *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, San Antonio, Texas, USA, 2000.

[11] Ö. Yilmazel, C. M. Finneran and E. D. Liddy, "Metaextract: an NLP system to automatically assign metadata," in *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, Tucson, AZ, USA, 2004.

[12] W. Paik, S. Yilmazel, E. Brown, M. Poulin, S. Dubon and C. Amice, "Applying Natural Language Processing (NLP) Based Metadata Extraction to Automatically Acquire User Preferences," in *Proceedings of the 1st International Conference on Knowledge Capture*, New York, NY, USA, 2001.

[13] P. Flynn, L. Zhou, K. Maly, S. Zeil and M. Zubair, "Automated Template-Based Metadata Extraction Architecture," in *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, Berlin, 2007.

[14] M. contributors, "HTML: HyperText Markup Language," Mozilla, 17 July 2023. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTML.

[15] "What is natural language processing (NLP)?," IBM, [Online]. Available: https://www.ibm.com/topics/natural-language-processing. [Accessed 07 08 2023].

[16] "A complete guide to Natural Language Processing," DeepLearning.AI, 11 01 2023. [Online]. Available: https://www.deeplearning.ai/resources/natural-language-processing/. [Accessed 07 08 2023].

List of Figures

Acknowledgments

I want to express my gratitude to everyone who has helped me over these years, especially in the last year. This thesis is the end of a long journey that I could not have finished without the help and guidance of many people.

A special thanks to Prof. Letizia Tanca and Dr. Eng. Davide Piantella for their guidance and support during this thesis.

I also want to thank my family, my girlfriend, and friends for always being there for me while I was working on this thesis. You all motivated me to finish this work.