I. J. Computer Network and Information Security, **2022, 6, 1-13**
Published Online on December 8, 2022 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2022.06.01

Dynamic Editing Distance-based Extracting Relevant Information Approach from Social Networks

Mohamed Nazih Omri*
MARS Research Laboratory LR17ES05, University of Sousse, Tunisia
E-mail: MohamedNazih.Omri@eniso.u-sousse.tn
ORCID iD: https://orcid.org/0000-0001-7803-0179
*Corresponding Author

Fethi Fkih
Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia
MARS Research Laboratory LR17ES05, University of Sousse, Tunisia
E-mail: f.fki@qu.edu.sa
ORCID iD: https://orcid.org/0000-0001-8937-9616

Received: 24 July 2022; Revised: 20 September 2022; Accepted: 14 October 2022; Published: 08 December 2022

Abstract: Online social networks, such as Facebook, Twitter, LinkedIn, etc., have grown exponentially in recent times with a large amount of information. These social networks have huge volumes of data especially in structured, textual, and unstructured forms which have often led to cyber-crimes like cyber terrorism, cyber bullying, etc., and extracting information from these data has now become a serious challenge in order to ensure the data safety. In this work, we propose a new, supervised approach for Information Extraction (IE) from Web resources based on remote dynamic editing, called EIDED. Our approach is part of the family of IE approaches based on masks extraction and is articulated around three algorithms: (i) a labeling algorithm, (ii) a learning and inference algorithm, and (iii) an extended edit distance algorithm. Our proposed approach is able to work even in the presence of anomalies in the tuples such as missing attributes, multivalued attributes, permutation of attributes, and in the structure of web pages. The experimental study, which we conducted, on a standard database of web pages, shows the performance of our EIDED approach compared to approaches based on the classic edit distance, and this with respect to the standard metrics recall coefficient, precision, and F1-measure.

Index Terms: Information Extraction, Mask Induction, Inductive Learning, Edit Distance, Alignment, Edit Operations.

1. Introduction

1.1. Context and Issues

The web has become one of the richest sources of information and a medium of choice for finding the answer to a need for information. On the web you can find various sources of useful information such as telephone directories, product catalogs, search engines, etc. However, if the answer to a given need can potentially be found on the Internet, the volume of information made available to users can make this task complex. This volume of data is both a problem (how to find the right information?) and also an opportunity because the redundancy makes it possible to automatically identify relevant sequences in the texts (in particular by learning methods) [1]. For example, to find the cheapest price of a product, it is then necessary to interrogate several different data sources. Then, in each source, you have to browse a list of results presented on several pages. Eventually, once the answers are found, they will need to be extracted and stored. In this introduction, we first present the working context. Then we focus on the problem. And we end with the report's organizational plan. Finding relevant information on the web that meets a user's query requires first locating that information on a web page, i.e. extracting relevant information from web pages. Nowadays, search engines only locate keywords on web pages. The purpose of Information Extraction (IE) is to structure information relevant to a particular domain. It involves identifying instances of an event or relationship class and extracting its attributes. The information is first extracted, to then be used to deduce more complex facts by inferences or by the resolution of

coreferences.

Since its appearance, the Web has not stopped evolving by integrating more and more services for customers who are becoming more and more numerous. The new generation of the web called the Semantic Web promises to provide an environment for translating the structure of web pages into meaningful content [2]. This will be possible through the work carried out in the field of IE. Therefore, Semantic Web would be a vast space for the exchange of resources between machines allowing the exploitation of large volumes of information and various services, helping users by freeing them from a (large) part of their research work, and a combination of these resources. This visionary proposal builds on the considerable success of the Internet. It also aims to take advantage of the growing digitization of collections and the generalization of the production of documents in a standardized format or structured using XML.

Today, the technological effort is made and the desire to have consensual resources is real: the W3C has set up several working groups which have defined standards compatible with XML. This choice is justified by the desire that the knowledge and information thus represented be associated with the documents as metadata, and treated in a homogeneous manner with the marks describing the structure of the data. The artificial intelligence and knowledge engineering scientific community as a whole has embraced this project and has embraced it with enthusiasm. Proof of this is the explosion in the number of works, conferences, and reviews on the Semantic Web since 1999.

1.2. Contribution

In the framework of this work, we propose a new approach, articulated around different modules, in order to ensure an efficient information extraction based on the dynamic edit distance from Web resources. The main contributions of our solution are summarized into the following points:

- Proposing a first Web page labeling module that allows the user to specify the architecture of a Web page. This module exploits the particularity of XML extended markup language to store the label of the Web page in an XML file.
- Developing a second inductive mask learning module to calculate the similarity threshold between user-specified records. It also allows you to create the information extraction rule based on the Costs matrix which defines the costs of elementary operations on tokens.
- Implementing a third inference module makes it possible to extract the relevant tuples from a web page while considering the learned parameters.

1.3. Paper Organisation

The rest of this article is organized as follows. In section 2, we review the classifications of IE approaches and we highlight IE approaches from web resources. Therefore, we introduce in section 3 our proposed approach for Information Extraction from Web pages based on Dynamic Editing Distance: EIDED. In section 4, we present the obtained results of the performance tests of this new approach and we provide an in-depth discussion. Finally, in section 5, we conclude by summarizing the proposed work and we give the main prospects for the proposed approach.

2. Related Works

In recent years, several IE approaches have been proposed. In [3-5] authors have proposed a Pertinence Feedback and Semantic Networks for Goal's Extraction as well as a Relevance Feedback for Goal's Extraction from Fuzzy Semantic Networks and A QoS-aware approach for discovering and selecting configurable IaaS Cloud services. The MUC (Message Understanding Conferences) data warehouse inspired early work in IE. The contribution of the MUC warehouse in the field of IE even leads researchers to classify IE approaches into two classes. The first class of MUC IE approaches: AutoSolg [6], LIEP [7], PALKA [8], etc. And the second class of IS approaches after MUC: WIEN [9], SoftMealy [10], WHISK [11], etc.

In [12], the authors proposed an approach for validating web information extractors. This approach relayed on applying performance metrics for measuring the effectiveness of web IE models. Ranjan et al. in [13], used IE models for generating a profile for a given person. In fact, they introduced a system that extract information about a person from available data on social media, and then rank the extracted information using Machine learning [13,14]. In the same context, Patnaik and Babu in [15] and Ramalingram et al. in [16] proposed very similar approaches based on deep learning models for information extraction using You Only Look Once and Long Short-term Memory (LSTM) networks in order to proactive failure prediction of the web pages' URL. These models were mainly proposed to overcome issues related to dynamic changes in web pages' layout.

As an application on the financial field, the authors in [17] used deep learning and improved Hidden Markov Models for extracting financial information from Web pages and they obtained good results. Moreover, Nair et al. in [18] proposed a novel IE approach applied on medical domain. In fact, they aimed to extract diseases symptoms from medical websites using NLP techniques and a medical vocabulary. In order to automatically extract structure information from web pages, the authors in [19] introduced WebFormer, a Web-page transFormer model that tries to solve the problem of the diversity of web layout patterns. For reaching the same purpose, the authors in [20] proposed to pre-train a transformer model (WebFormer) according to a set of four objectives based on the structure of the web

documents.

The purpose of the induction mask is to automatically produce a mask that is used to extract relevant information for an information source. We can classify induction mask approaches into four categories: Manual IE approaches, Supervised IE approaches, Semi-supervised IE approaches, Automatic IE approaches. In the following section, we outline for each class the IE approaches most cited in the literature.

2.1. Supervised IE Approaches

Supervised mask induction approaches accept as input a set of indexed web pages. These approaches make it possible to flip an induction mask. The user first provides a set of tagged pages and the approach can suggest which pages to add up for the user to index via the GUI. In what follows, we mainly highlight approaches to IE that use the construction of masks in a supervised way.

A. WIEN (Environment Induction Wrapper)

In this work on the induction of masks [20,21,9], Kushmerick et al. have, on the one hand, proposed a first formalization of the mask construction problem for a Web information source and, on the other hand, have implemented a mask induction system: the WIEN system. The formalization of Kushmerick et al. gave rise to a first method of inducing masks from indexed example pages. However, the performance of the algorithms can be improved by reducing the candidate space with a finer analysis of the constraints.

B. SoftMealy

The SoftMealy [10] mask induction approach mainly tries to provide a solution to the problems of attribute ordering and missing attributes. The extraction rules that he seeks to build must take into account the different permutations of the attributes that appear in the occurrences of the relation to be extracted. The representation of extraction masks by SoftMealy is based on finite state automata FST (Finite State Transducer). Moreover, SoftMealy is no longer based on delimiters but on separators. A separator is used to characterize a position both from the text located just before this position and from the text located just after. A separator then takes into account both what is to the left and to the right of the position it determines. This position corresponds either to the beginning or to the end of a value. Thus, even the format of the content of this value is taken into account by the separator.

2.2. Semi-supervised IE Approaches

As opposed to supervised IE approaches, semi-supervised approaches accept an approximate example introduced by users for the extraction mask generation. Semi-supervised IE approaches do not require any indexed pages for training. This minimizes human intervention.

A. IEPAD (Information Extraction Based on Pattern Discovery)

IEPAD is a semi-supervised IE approach that requires user intervention to choose from the generated patterns and correct the patterns so that they can fully extract all values without extracting too many. The choice of patterns is not always easy because the user does not necessarily have the knowledge necessary to understand the patterns. Moreover, it is not necessarily easy for the user to understand the effects of the different parameters (the thresholds for each of the measurements) on the extraction. Furthermore, the implicit assumption that the textual parts are indivisible is problematic. Indeed, it happens that several pieces of information of different natures that need to be separated are found in the same textual part.

B. Automatic IE Approaches

Unsupervised IE approaches do not employ any example indexed pages and do not require any human intervention to produce an extraction mask. Unlike supervised IE approaches where data mining is specified by users. For automatic IE approaches data extraction is defined as data that is used to produce in-page text or unindexed text in regions of the input page [22]. In some cases, several schemas may conform to the learned pages due to the presence of irrelevant attributes, leading to ambiguity [23]. The choice for determining the correct scheme is left to the users. On the contrary, if all the attributes are relevant, only one schema is necessary for the extraction. Automated IE approaches include RoadRunner, MDR (Mining Data Records), DEPTA (Data Extraction Based on Partial Tree Alignment), GCNTree (Graph Convolutional Model on Tree Structure) [24], TBPM (Tree-Based Pattern Matches) [25], Network Topology Coincidence Degree [26], etc.

3. Proposed IE Approach based on Dynamic Editing Distance

We present in this section our approach to handle the problem of IE from Web pages. First, we emphasize our preliminary idea. Then, we discuss the general architecture of our EIDED approach and the block diagram. After that, we present the different EIDED algorithms. Then, we explain our approach with an illustrative example. And finally, we end with the study and the results of the experiment. Our approach is part of the family of IE approaches based on an extraction mask with anomaly management. Our EIDED approach is supervised and It is composed of two

algorithms: a learning algorithm and an inference algorithm. Furthermore, it is based on the extended edit distance algorithm.

In our approach, we have defined a web page as a sequence of tokens. Each token represents the basic unit of information in a web page and can take the form of a single character, an HTML tag, a string, etc. We have used the term view and the typographic view. In our approach, we deal with anomalies that can occur in web pages. EIDED consists of three core modules: Web page labeling module, Learning module and Inference module.

3.1. Architecture of a Web Page

We based our definition of the structure of Web pages on that used by SoftMealy [10]. We have noticed that a web page is made up of a set of web objects. Each web object is made up of a record. A record contains a set of attributes. Therefore, a web page consists of two main areas: record area and attribute area. The record area represents the record to extract. The attribute zone represents the text fragment to be extracted contained in the record. We will call record, the tuple to extract and the attributes of styles which include the attributes.

3.2. Token Classes

A web page can be defined as a sequence of tokens where each token belongs to a class of tokens. In the following table 1, we present the alphabet representing the twelve classes of tokens that we will use with their examples:

Table 1. Description of different used classes.

Class	Sub-class	Description
Class 1	C1	Uppercase character string:
	C2	A string starting with an uppercase letter and followed by at least one lowercase
	C3	A string starting with a lowercase letter and followed by zero or more characters: .
	C4	A numeric string: .
	C7	Punctuation symbol: .
	C12	A generic string representing any class of the classes already listed.
Class 2	C5	An opening HTML tag: .
	C6	A closing HTML tag: .
	C8	An opening HTML tag representing a control character: , , and , etc.
	C9	A closing HTML tag representing a control character.
	C10	An opening HTML tag representing an element of a list: .
	C11	A closing HTML tag representing an element of a list: .

3.3. General Architecture of EIDED

EIDED is built around three core modules: a module for labeling training examples, a module for training examples, and an inference module for extracting relevant tuples. The web page labeling module receives as input a web page and the schema. The schema specifies the number of relevant fields and their respective names. It is given by the user. The label module is used to build the training examples. By using this module, the user indexes relevant information in a web page. The indexed web pages will serve as inputs for the learning module in order to learn the extraction mask (IE rule). Our inference module aims to extract the different tuples contained in a web page based on the extraction mask obtained at the output of the learning module.

3.4. Different Basic Modules of EIDED

Our approach can be summarized in three modules. In what follows, we will detail each of these modules separately.

A. Web Page Labeling Module

As described in algorithm 1, this module allows you to specify the architecture of a Web page. It specifies the beginning and the end of each zone in the page. The web page labeling module receives as input a web page and a schema. It allows to index the relevant fields in the page. The web page labeling algorithm allows the user to specify the boundaries of different areas within the page. Algorithm 1 takes as input a web page and its schema and returns the indexed page.

Algorithm 1. Web page labeling algorithm.

```
Algorithm Label
Input : web page, its schema
Output : Label of a web page
1. Begin
2.          L<—[]
3.          Specify the n Ei records
4.          L<—[E1, E2, …En]
5.          For each Ei in [E1, E2, …En] do
6.                Specify the k attributes Ai, 1<i<=k of the record Ei
7.          Ei<—[A1, A2, …Ak]
8.          end
9.    return(L)
10.   End.
```

B. Learning Module

The indexed web pages obtained by the user using the web page labeling module will serve as inputs for the learning module. The learning algorithm determines the extraction mask of the indexed web pages. This mask is defined by the set of indexed "record" zones where each token of the record is generalized using one of the twelve classes of tokens already defined. To deal with anomalies that may exist in the records, we calculate the degree of similarity between the different identified records using the edit distance. To determine the degree of similarity between records, we defined the following Similarity function:

$$\boldsymbol{Similarity}: \mathbb{N} \times \mathbb{N}: \rightarrow [\mathbf{0}, \mathbf{1}]$$

$$Similarity(E_1, E_2) = \frac{1}{1+Editingdistance(E_1,E_2)} \quad (1)$$

$Where: E_1, E_2 are\ two\ records.$

We have seen that the edit distance already defined in section 2 measures the number of elementary operations to go from one string to another. An elementary operation is an edit operation on the tokens to transform one sequence of tokens into another. We have four basic operations:

- Inserting a new token from Sequence1 into Sequence2.
- Deleting a token from Sequence1.
- Substitution of a Sequence1 token with a Sequence2 token.
- Equality of a Sequence1 token with a Sequence2 token.

For two records that have an editing distance of 0; this means that it is the same recording. The degree of similarity between these two records is equal 1 since the edit distance is equal 0. Moreover, for two more and more different recordings, the editing distance increases. We will have the degree of similarity between these two records weakens.

Our learning algorithm 2 accepts pairs of web pages and their respective labels as input. It returns an extraction mask. This mask is formed by the record structures specified by the user during the web page labeling module, the similarity threshold and the cost matrix. Below is the statement of the learning algorithm 2.

In our learning algorithm 2, we set the similarity threshold as follows:

$$Threshold\_Sim = \max_{E_1,E_2}(Similarity(E_1, E_2)) \quad (2)$$

This equality indicates that the similarity threshold is set to the maximum value of the Similarity function. The purpose of this choice is to promote precision despite the recall coefficient. As an indication, the similarity threshold can be chosen at the minimum value of the similarity function. In this case, the recall coefficient is favored in spite of the precision. Learning the extraction mask is done in three phases: (i) Learn about different record structures, (ii) Learning the similarity threshold. And (iii) Learning the costs of token operations.

Algorithm 2. Learning Algorithm

Algorithm for Learning
Input: {(PageWeb, Label)}
Output: Extraction mask (Learned Record Structure, Similarity Threshold, Cost Matrix)
Begin
For each couple of web pages **do**
    For each pair of records (Ei, Ej) in the Web pages **do**
        Calculate editing distance: Editing Distance(Ei, Ej)
        Calculate degree of similarity: Similarity(Ei,Ej)=
        Determine alignments: Alignment
    End
End
Determine the different structures of learned records: Learned Record Structure
Determine the similarity threshold: Threshold_Sim=max(Similarity)
For all recordings **do**
    Build the cost matrix: Matrix_Costs
End
Return Extraction Mask(Struct_Energ_Learned, Threshold_Sim, Matrix_Costs)
End.

a. Learning Record Structures and Similarity Threshold

For the inference process to extract the different records, the learning module must determine the different structures of the records seen in the learned web pages. The second step in learning the extraction mask is to determine the degree of similarity between the learned records. This step makes it possible to set an authorized error threshold in order to affirm that a tuple is relevant.

b. Learning About Costs

Cost learning serves to make the classical edit distance algorithm dependent on the learned examples since it allows to determine the costs based on the learned examples. The classic edit distance algorithm is based on static costs. Indeed, the cost of an operation on a token is equal 0 or is equal 1. We intend to use dynamic costs to take into account the learned examples. Additionally, an anomaly that may occur in the records of a web page may take one of the following forms:

- Inserting a new token from Record1 into Record 2.
- Deleting a token from Record 1.
- Substitution of a token from Record1 with a token from Record 2.
- Equality of a token from Record1 with a token from Record 2.

Based on this formalization, the cost learning phase can be divided into two tasks. The first task is to determine the different alignments between the different learned Web page pair records. The second task consists in calculating the cost of an elementary operation on a token while considering the alignments. We recall that we have four elementary operations:

- Inserting a new token from Sequence1 into Sequence 2.
- Deleting a token from Sequence1.
- Substitution of a Sequence1 token with a Sequence 2 token.
- Equality of a Sequence1 token with a Sequence 2 token.

To define the costs of operations on the tokens, we will draw up the cost matrix which determines the degree of contribution of each token in the calculation of the degree of similarity. Equation 3 shows the matrix structure:

$$Costs = \begin{matrix} token1 & token2 & \dots & token12 \\ \begin{pmatrix} Cost_{1,1} & Cost_{1,2} & \dots & Cost_{1,12} \\ Cost_{2,1} & Cost_{2,2} & \dots & Cost_{2,12} \\ Cost_{3,1} & Cost_{3,2} & \dots & Cost_{3,12} \end{pmatrix} & \begin{matrix} Insertion \\ Removal \\ Substitution \end{matrix} \end{matrix} \quad (3)$$

Where, Cost: determines the value of the cost of an operation on a token. To calculate the costs of token operations, we determine in the alignments seen during training the costs by the following function (equation 4):

$Where, O: the\ set\ of\ elementary\ operations$
$J: the\ set\ of\ tokens$

$$Cost: \mathbb{N} \times \mathbb{N} \rightarrow [0,1]$$

$Cost\ computes\ the\ appearance\ degree\ of\ the\ operation\ of\ the\ token\ in\ the\ alignments,$
$defined\ as\ follows$:

$$\begin{cases} Cost(o,j) = 0 \quad if\ o = equqlity\ of\ two\ jetons \\ \qquad Cost(o,j) = 1 - \frac{x}{y}, otherwise \end{cases} \tag{4}$$

$Where: o \in O; j \in J$
$x: Appearance\ number\ of\ the\ operation\ of\ the\ token\ in\ the\ alignments$
$y: Alignments\ number$

In fact, the function ***cost*** determines the degree of appearance of the token operation in the alignments inferred from learned web pages. In other words, for an operation on a token seen in the learned alignments, we try to reduce the value of its cost compared to the static cost which is equal to 0. The function cost can delegate its work to four other functions:

- $Cost_{insertion}$ for the cost of the elementary operation of token insertion.
- $Cost_{removal}$ for the cost of the elementary operation of token removal.
- $Cost_{substitution}$ for the cost of the elementary operation of token substitution.

C. Inference Module

Our inference module aims to extract the different tuples based on the extraction mask obtained from the learning module. To recognize that a tuple is relevant, one must calculate the degree of similarity between the sequence of tokens and the structures of the learned records while considering the costs of the learned operations. Our inference algorithm3 allows to extract the tuples contained in a web page. In the following, we present the statement of the inference algorithm3.

Algorithm 3. Algorithm of inference.

Algorithm of inference
Input: Web page
Extraction mask
Begin
1. **For** the selected record E_i **from** E_1 to E_n **do**
2. ***Define***: E_i the length of selected record
3. ***Define***: F *Sliding window of length* n
4. **Calculate**: the dynamic editing distance:
5. $\text{Dynamic\_Editing\_Distance}(F, E_i) = \text{Dynamic\_Editing\_Distance}(F, E_i)$
6. **Calculate** the similarity degree:
7. $\text{Similarity}(F, E_i) = \frac{1}{1 + \text{Dynamic\_Editing\_Distance}(F, E_i)}$
8. **If** $\text{Similarity}(F, E_i) \geq Similarity_{threshold}$ **Then**
9. F is a relevant tuple
10. **End If**
11. Move the window F forward
12. **End do**
End

In the inference algorithm 3, we fixed the length of the window F as follows:

$$WindowLengthF = learnedrecordlengthEi \tag{5}$$

This condition indicates that the length of the window F is equal to the length of the learned record. The extraction of a sequence of tokens is done based on the learned record. Hence the need to set the size of the sliding window to the length of the learned record. We have set the condition for extracting a relevant tuple as follows:

$$Similarity >= Similarity_{threshold} \tag{6}$$

This condition makes it possible to authorize a margin of error on the similarity which is calculated based on the dynamic editing distance algorithm greater than a certain learned threshold. To extract the different tuples found in a web page, the inference algorithm 3 uses Algorithm 4 to determine the dynamic edit distance between the window and the learned records.

Algorithm 4. Dynamic editing distance algorithm.

```
Algorithm Dynamic Editing Distance
Input :   String Sequence 1, String Sequence 2
          Declare a matrix of integers [0…SL1, 0…SL2]
          /*SL 1 is Sequence length 1, SL2 is Sequence length 2*/
          Declare the integers i, j, Cost
Output : d[LS1, LS2]  /* the dynamic editing distance between two strings*/
Begin
1.   For i from 0 to SL1 do
2.        d[i, 0]
3.   end do
4.   For j from 0 to SL2 do
5.        d[i, 0]
6.   end do
7.   For  i from 1 to SL1 do
8.        For j from 1 to SL2 do
9.             If chain1[i-1]=chain2[j-1] then
10.                 Cost <— 0
11.            else
12             Cost <— token substitution cost
13.            d[i, j] <— min(d[i-1, j]+token insertion cost, d[i, j-1]+token removal cost, d[i-1, j-1]+ cost)
14.       End if
15.  End do
16. End do
End.
```

4. Experimentation and Analysis of Results

In this section, we carry out an experimental comparison of our proposed models and 2 other well-known models in the literature. We have to mention that the experimental study was performed on an intel (R) Core™ *i7* machine with a clock frequency of 2.3 Ghz and 16 GB of RAM running Windows 10 and JavaScript programming language for implementing all the proposed algorithms.

4.1. Test Data

We worked on the standard basis of web pages describing the official telephone codes of different countries. We tried to compare the performance of our EIDED approach with that of the IE approach based on the classic edit distance and the SoftMealy approach. We considered five collections of web pages that exhibit different types of anomalies. We have tried to extract for each web page the different tuples it contains. The results are summarized in Table 2.

4.2. Testing Metrics

To test the performance of EIDED, we tried to compare it with IE's approach based on the classic edit distance and with the SoftMealy approach based on the three standard measures: recall, precision, and the F1-measure.

The recall is defined by the number of tuples retrieved and relevant by the total number of tuples on the page:

$$Recall = \frac{NTPE}{NTTP} \tag{7}$$

Where NTPE represents the number of relevant tuples extracted and NTTP is the total number of tuples in the Web page.

Accuracy is defined by the number of tuples extracted and relevant by the number of tuples extracted:

$$Precision = \frac{NTPE}{NTE} \tag{8}$$

Where: NTE is the number of tuples extracted.

The F1-measure is defined by the product of recall and precision multiplied by two by the sum of recall and precision:

$$F1_{measure} = \frac{2*Recall*Precision}{Recall+Precision} \quad (9)$$

4.3. Experimentation and Analysis of Results

We have represented the comparative curves of three evaluation metrics recall, precision and F1-measurement in the following three figures: 2, 3 and 4.

Table 2. Table summarizing the results obtained after simulation.

	Set of Web page	S0 (11pages)	S1 (22pages)	S2 (37pages)	S3 (45pages)	S4 (69pages)
	total number of tuples	37	70	122	143	228
Proposed approach EIDED	Number of tuples extracted	45	33	58	71	111
	Number of pertinent tuples extracted	32	25	46	55	84
IE based on classical Euclidean distances	Number of tuples extracted	32	20	34	36	58
	Number of pertinent tuples extracted	23	15	27	25	46
SoftMealy	Number of tuples extracted	27	13	32	29	45
	Number of pertinent tuples extracted	8	11	20	26	45

We have represented the results obtained in table 2 on a histogram in Figure 1. For each set of web pages, the left bar represents the number of tuples extracted (NTE) for each approach and the right bar represents the number of relevant tuples extracted (NTPE).

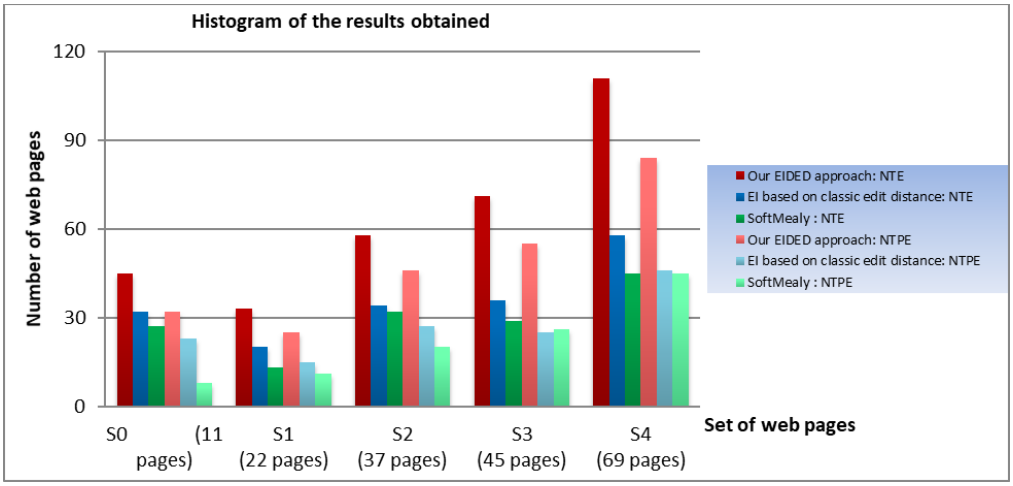

Fig.1. Histogram of the results obtained.

We note that the number of tuples extracted by our EIDED approach is higher than that by the IE approach based on the classic edit distance since we exploited the notion of dynamic costs. Moreover, the number of tuples extracted by EIDED is higher than that of the SoftMealy approach because this approach is based on deterministic automata; it does not allow anomalies in the tuples. The effectiveness of our approach increases further when the pages' number increases from one set to another. In our inference algorithm, we chose an error threshold to favor precision despite the recall. This choice can be read in figures 2 and 3.

We notice that the three curves follow the same shape because the learning instances are the same for the three approaches. Moreover, there is a fall in the shape of the three curves since S0 is the learned set. Figure 3 clearly illustrates that the recall of EIDED is always superior to the recall achieved by the other two approaches: classical edit distance-based IE and SoftMealy. The gap between EIDED and the other two approaches grows as the number of web pages increases.

We notice that the accuracy of our EIDED approach is generally superior to that of the classical edit distance-based IE approach and the SoftMealy approach. Moreover, SoftMealy has a very varied appearance according to the

number of Web pages; which is not desirable. On the contrary, EIDED has almost the same constant pace. The accuracy of EIDED is slightly better than the classic edit distance-based approach of IE.

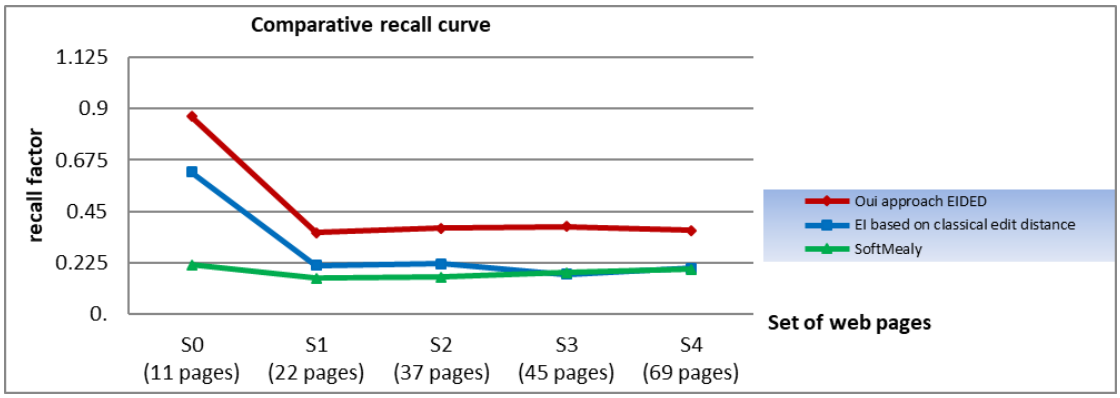

Fig.2. Comparative recall curve.

The F1-measure is used to measure the performance of an IE approach. It combines recall and precision in a single expression. We notice that the shape of the curve of the SoftMealy approach does not exceed the value. On the contrary, the curves of our EIDED approach and of the IE approach based on the classic edit distance look almost the same. Figure 4 clearly illustrates that the F1-measure of EIDED is always higher than that achieved by the other two approaches. Moreover, the F1-measure of EIDED has no value less than. This explains the performance of our EIDED approach compared to the other two approaches.

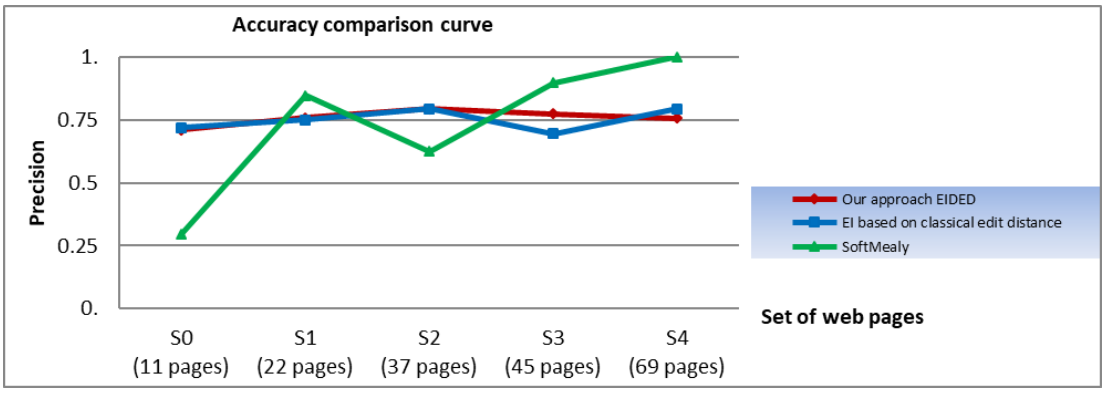

Fig.3. Accuracy comparison curve

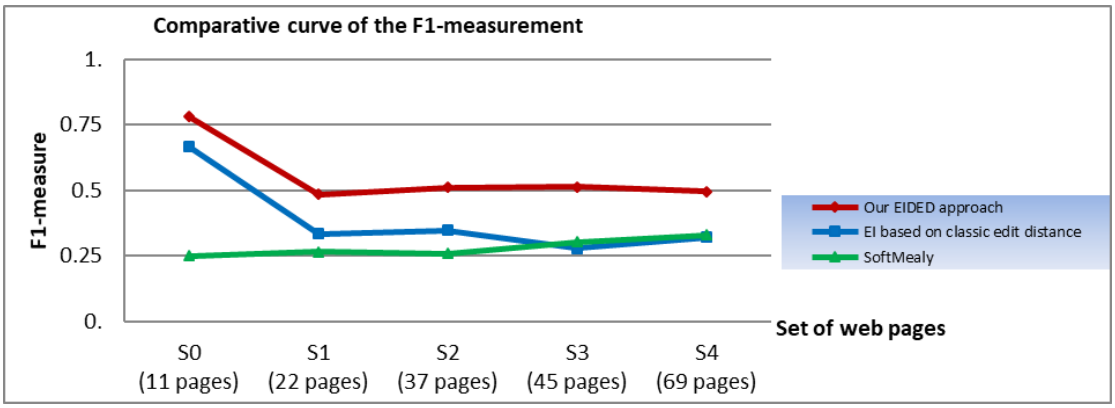

Fig.4. Comparative curve of the F1-measurement

4.4. Discussion

We have proposed a new approach to relevant IEs for semi-structured web pages. Our EIDED approach is able to detect several anomalies at the same time, for example, missing attributes, attribute permutations, etc. This is made possible by exploiting the architecture of a web page and the dynamic edit distance algorithm.

Our EIDED approach requires few learning examples compared to the other approaches tested. When the trained web pages represent the hypothesis sought, the number of training examples tends to decrease. In some cases, a single

training example is enough to achieve good performance. The user can intervene at any time to improve the learned rules by adding new learning examples. The learning and inference modules are independent of the lexical analyzer used. Indeed, the tokens are replaced in the learning and inference process by the numbers of the respective class.

IE rules are based on token transaction costs. These rules play the role on the one hand for the IE since it serves to measure the degree of similarity between the records and on the other hand for the definition of the degree of token contribution in the calculation of the dynamic edit distance.

Experimental results obtained on several test web pages show the superior performance of our EIDED approach compared to that of the classic edit distance-based IE approach and the SoftMealy approach with respect to all three metrics: precision and F1-measurement. Moreover, EIDED's F1-measurement keeps a constant value for an increasing number of Web pages. This is a good performance indicator of our EIDED approach since the number of web pages on the Internet is constantly increasing.

Our proposed model can be used in many domains related to Information Retrieval and Semantic Web fields [27], such as, Web document indexing [28], Author Profiling [29], Social Network safety and security [30], etc.

5. Conclusion and Prospects

5.1. Summary

In this work, we reviewed web-based IE approaches by classifying them into four classes: manual, semi-supervised, supervised, and automatic information retrieval approaches. Based on this study, we developed our EIDED approach for relevant IEs from web resources by learning inductive masks. Our EIDED approach is able to work even in the presence of anomalies in the records and in the structure of a web page. The definition of the structure of a Web page, allowed us to develop a simple learning algorithm and an efficient inference algorithm while considering the notion of dynamic editing distance. We tested and compared our EIDED approach with two other SoftMealy approaches and the classic IE approach based on edit distance on a standard web page basis. Overall, EIDED showed several advantages over these two approaches. We noticed that EIDED is more efficient since the recall and the F1 measure are always higher than the other models. Regarding Accuracy, our model has been outscored by SoftMealy when the number of web pages' set is greater to 53, but it remains stable during the test and provides a good accuracy for all the page's sets, in opposite to the other models. This good finding can be explained by the fact that our proposed model can extract more relevant tuples than the other two approaches which increases the recall and the accuracy and, subsequently, increases the F measure.

5.2. Prospects

Although the extraction masks used by our inference algorithm efficiently extract tuples from a web page, nevertheless, these masks become inefficient when the general structure of the information source changes. Our first direction is therefore to solve this problem by developing another mask reconstruction module based on the results provided by the inference module. Users express their requests through the specification of the Web page, but our EIDED approach does not allow the latter to specify the attribute he needs. Our second direction is to improve the inference module so that the user can express the relevant attributes he needs. The third direction is to conduct a more in-depth comparative study between the main approaches studied in the literature in order to give more amplification to scholars and practitioners on how to effectively extract information from web resources.

References

[1] Asma Omri, Mohamed Nazih Omri, "Towards an Efficient Big Data Indexing Approach under an Uncertain Environment", International Journal of Intelligent Systems and Applications, Vol.14, No.2, pp.1-13, 2022.

[2] R.Umagandhi, A.V. Senthil Kumar,"Evaluation of Reranked Recommended Queries in Web Information Retrieval using NDCG and CV", International Journal of Information Technology and Computer Science, vol.7, no.8, pp.23-30, 2015.

[3] Mohamed Nazih Omri. Possibilistic Pertinence Feedback and Semantic Networks for Goal's Extraction. Asian Journal of Information Technology (AJIT) 3 (4), 258-265. 2004.

[4] Mohamed Nazih Omri. Relevance Feedback for Goal's Extraction from Fuzzy Semantic Networks. Asian Journal of Information Technology (AJIT). 3 (6), 434-440. 2004.

[5] Jalel Eddine Hajlaoui, Mohamed Nazih Omri, Djamal Benslimane. A QoS-aware approach for discovering and selecting configurable IaaS Cloud services. Computer Systems Science and Engineering 32 (4). 2017.

[6] Ranjan, R., Vathsala, H. & Koolagudi, S.G. Profile generation from web sources: an information extraction system. Soc. Netw. Anal. Min. 12, 2. 2022. https://doi.org/10.1007/s13278-021-00827-y

[7] Nicholas Kushmerick. Wrapper Induction for Information Extraction. PhD thesis, University of Washington, 1997.

[8] Nicholas Kushmerick, Daniel S. Weld and Robert Doorenbos. Wrapper Induction for Information Extraction. Proceedings of the Fifteenth International Conference on Artificial Intelligence (IJCAI), pp. 729-735, 1997.

[9] Kejriwal, M. Information Extraction. In: Domain-Specific Knowledge Graph Construction. SpringerBriefs in Computer Science. Springer, Cham. 2019. https://doi.org/10.1007/978-3-030-12375-8_2.

[10] Chia-Hui Chang, M. Kayed, M. R. Girgis and K. F. Shaalan, "A Survey of Web Information Extraction Systems," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1411-1428, Oct. 2006, doi: 10.1109/TKDE.2006.152.
[11] Yu Guo, Zhengyi Ma, Jiaxin Mao, Hongjin Qian, Xinyu Zhang, Hao Jiang, Zhao Cao, and Zhicheng Dou. 2022. Webformer: Pre-training with Web Pages for Information Retrieval. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22). Association for Computing Machinery, New York, NY, USA, 1502–1512. https://doi.org/10.1145/3477495.3532086.
[12] Patricia Jiménez, Rafael Corchuelo, On validating web information extraction proposals, Expert Systems with Applications, Volume 199, 2022, 116700, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2022.116700.
[13] Leila Helali, Mohamed Nazih Omri, "Heuristic-based Approach for Dynamic Consolidation of Software Licenses in Cloud Data Centers", International Journal of Intelligent Systems and Applications, Vol.13, No.6, pp.1-12, 2021.
[14] Mohamed Nazih Omri, Wafa Mribah, "Towards an Intelligent Machine Learning-based Business Approach", International Journal of Intelligent Systems and Applications, Vol.14, No.1, pp.1-23, 2022.
[15] Sudhir Kumar Patnaik and C. Narendra Babu. 2022. A Web Information Extraction Framework with Adaptive and Failure Prediction Feature. J. Data and Information Quality 14, 2, Article 12 (June 2022), 21 pages. https://doi.org/10.1145/3495008
[16] M. Ramalingam, D. Saranya, R. ShankarRam, P. Chinnasamy, K. Ramprathap and A. Kalaiarasi, "An Automated Framework for Dynamic Web Information Retrieval Using Deep Learning," 2022 International Conference on Computer Communication and Informatics (ICCCI), 2022, pp. 1-6, doi: 10.1109/ICCCI54379.2022.9741044.
[17] Ping Yang (2022) Financial Information Extraction Using the Improved Hidden Markov Model and Deep Learning, IETE Journal of Research, DOI: 10.1080/03772063.2022.2054873
[18] Nair, P.C., Gupta, D., Indira Devi, B. Automatic Symptom Extraction from Unstructured Web Data for Designing Healthcare Systems. In: Shetty, N.R., Patnaik, L.M., Nagaraj, H.C., Hamsavath, P.N., Nalini, N. (eds) Emerging Research in Computing, Information, Communication and Applications. Lecture Notes in Electrical Engineering, vol 790. 2022. Springer, Singapore. https://doi.org/10.1007/978-981-16-1342-5_46
[19] Qifan Wang, Yi Fang, Anirudh Ravula, Fuli Feng, Xiaojun Quan, and Dongfang Liu. 2022. WebFormer: The Web-page Transformer for Structure Information Extraction. In Proceedings of the ACM Web Conference 2022 (WWW '22). Association for Computing Machinery, New York, NY, USA, 3124–3133. https://doi.org/10.1145/3485447.3512032
[20] Rinaldo Lima, Bernard Espinasse, and Fred Freitas. 2010. An adaptive information extraction system based on wrapper induction with POS tagging. In Proceedings of the 2010 ACM Symposium on Applied Computing (SAC '10). Association for Computing Machinery, New York, NY, USA, 1815–1820. https://doi.org/10.1145/1774088.1774471.
[21] Mirończuk, M.M. (2018). The BigGrams: the semi-supervised information extraction system from HTML: an improvement in the wrapper induction. Knowl Inf Syst 54, 711–776. https://doi.org/10.1007/s10115-017-1097-2.
[22] Fethi Fkih and Mohamed Nazih Omri. Estimation of a Priori Decision Threshold for Collocations Extraction: An Empirical Study. International Journal of Information Technology and Web Engineering (IJITWE), 8(3), 2013.
[23] Anupama Gupta, Imon Banerjee, Daniel L. Rubin. Automatic information extraction from unstructured mammography reports using distributed semantics. Vol 78, 78-86. 2018.
[24] Shuo Yang, Jingzhi Guo, Improved strategies of relation extraction based on graph convolutional model on tree structure for web information processing, Journal of Industrial Information Integration, Volume 25, 2022, 100301, ISSN 2452-414X, https://doi.org/10.1016/j.jii.2021.100301
[25] B. Bazeer Ahamed, D. Yuvaraj, S. Shitharth, Olfat M. Mirza, Aisha Alsobhi, Ayman Yafoz, "An Efficient Mechanism for Deep Web Data Extraction Based on Tree-Structured Web Pattern Matching", Wireless Communications and Mobile Computing, vol. 2022, Article ID 6335201, 10 pages, 2022. https://doi.org/10.1155/2022/6335201
[26] Zhinian Shu, Xiaorong Li, "Automatic Extraction of Web Page Text Information Based on Network Topology Coincidence Degree", Wireless Communications and Mobile Computing, vol. 2022, Article ID 9220661, 10 pages, 2022. https://doi.org/10.1155/2022/9220661
[27] Fethi Fkih and Mohamed Nazih Omri. Information Retrieval from Unstructured Web Text Document Based on Automatic Learning of the Threshold. International Journal of Information Retrieval Research (IJIRR), 2(4), 2012.
[28] Fethi Fkih and Mohamed Nazih Omri. Hybridization of an Index Based on Concept Lattice with a Terminology Extraction Model for Semantic Information Retrieval Guided by WordNet. In: Abraham, A., Haqiq, A., Alimi, A., Mezzour, G., Rokbani, N., Muda, A. (eds) Proceedings of the 16th International Conference on Hybrid Intelligent Systems (HIS 2016). HIS 2016. Advances in Intelligent Systems and Computing, vol 552. 2017. Springer, Cham. https://doi.org/10.1007/978-3-319-52941-7_15
[29] Sarra Ouni, Fethi Fkih and Mohamed Nazih Omri. Toward a new approach to author profiling based on the extraction of statistical features. Soc. Netw. Anal. Min. 11, 59 (2021). https://doi.org/10.1007/s13278-021-00768-6
[30] Duy Dang-Pham, Karlheinz Kautz, Ai-Phuong Hoang and Siddhi Pittayachawan. Identifying information security opinion leaders in organizations: Insights from the theory of social power bases and social network analysis. Computers & Security, Volume 112,2022.

Authors' Profiles

Mohamed Nazih Omri received his Ph.D. in Computer Science from the University of Jussieu, Paris, France, in 1994. He is a professor of computer science at the University of Sousse, Tunisia. Since January 2011, he is a member of MARS (Modeling of Automated Reasoning Systems) Research Laboratory. His group conducts research on Information Retrieval, Data Base, Knowledge Base, and Web Services. He supervised more than 20 Ph.D. and MSc students in different fields of computer science. He is a reviewer of many international journals such as Information Fusion journal, Psihologija Journal, and many International Conferences such as AMIA,

ICNC-FSKD, AMAI, SOMeT, etc.

Fethi Fkih received his Ph.D. in Computer Science from Faculty of Economics and Management of Sfax, Tunisia, in 2016. He is a member of MARS Research Laboratory at the University of Sousse, Tunisia. He is currently working as an assistant professor in the College of Computer, Qassim University, Saudi Arabia. His research interests focus on Artificial Intelligence, Text Mining, NLP, Recommender System, Web Mining, Sentiment Analysis, Information Retrieval, Document Indexing and Semantic Web.

How to cite this paper: Mohamed Nazih Omri, Fethi Fkih, "Dynamic Editing Distance-based Extracting Relevant Information Approach from Social Networks", International Journal of Computer Network and Information Security(IJCNIS), Vol.14, No.6, pp.1-13, 2022. DOI:10.5815/ijcnis.2022.06.01