# RULIE: Rule Unification for Learning Information Extraction

**Alexiei Dingli**
University of Malta
Malta
alexiei.dingli@um.edu.mt

**Dale P. Busuttil**
University of Malta
Malta
dbus0007@um.edu.mt

**Dylan Seychell**
University of Malta
Malta
info@dylanseychell.eu

## Abstract

In this paper we are presenting RULIE (Rule Unification for Learning Information Extraction), an adaptive information extraction algorithm which works by employing a hybrid technique of Rule Learning and Rule Unification in order to extract relevant information from all types of documents which can be found and used in the semantic web. This algorithm combines the techniques of the $LP^2$ and the BWI algorithms for improved performance. In this paper we are also presenting the experimental results of this algorithm and respective details of evaluation. This evaluation compares RULIE to other information extraction algorithms based on their respective performance measurements and in almost all cases RULIE outruns the other algorithms which are namely: $LP^2$, BWI, RAPIER, SRV and WHISK. This technique would aid current techniques of linked data which would eventually lead to fullier realisation of the semantic web.

## 1 Introduction

Information Retrieval and Extraction is a major area of interest in the field of computer science and the study of intelligent systems. Besides having various sources of information, we always strive to optimise our use of this information and devise new methods of how to understand this information and access it efficiently.

Search Engines retrieve information by using techniques based on keywords which map documents. However, the results returned by the search engines are usually various documents with most of them containing irrelevant information and therefore few documents which contain the information that is needed by the user.

In designing RULIE we kept in mind the fact that to help intelligent agents understand what is written in the page, the web pages need to be annotated [Berners-Lee and Fischetti, 1999] [Berners-Lee, 2001]. Annotations are metadata that are attached to pieces of text which can be used to give meaning to the content of the page. RULIE is an information extraction algorithm which can annotate data in a semi-automatic way thus relieving humans from doing so. In order to do this, such an algorithm must have rules upon which to act in order to make annotations feasible. These rules can be used to populate an ontology either automatically or manually through the use of a learning algorithm.

In the first part of this paper we will explain the fundamental concepts of information extraction together with a brief background of techniques used to enhance the results of information extraction engines. We will also give an overview of the types of texts which are handled by such algorithms. Subsequently we will briefly cover basic fundamentals of information extraction algorithms and analyse in details the $LP^2$ and BWI algorithms. The foundations of these two algorithms are used as a base for RULIE. They were chosen because they produce the best results overall as per [Department and Ciravegna, 2001]. Later in this paper we will present the design including the algorithm of RULIE and subsequently its evaluation. The positive results of RULIE are compared with other existing algorithms and we conclude this paper by presenting possible future directions for RULIE.

## 2 Information Extraction

Moens [Moens, 2006] defines Information Extraction as *"the process of selectively structuring and combining data that are explicitly stated or implied in one or more natural language documents"*. Information Extraction engines are built on two key principles which are mainly the Identification of Relevant Data and the Storing of relevant extracted data in Appropriate Structures[Téllez-Valero *et al.*, 2005].

The architecture of an information extraction system is normally composed of two key elements [Siefkes and Siniakov, 2005]. These are namely the *Learning Phase* which is a set of steps that will use part of the Training Corpus to build the underlying extraction model. The *Testing Phase* involves using the model produced by the learning phase with the test corpus to extract information on unseen cases and thus evaluate the model itself. [Siefkes and Siniakov, 2005].Sometimes Information Extraction systems also involve two other phases: the Pre-Processing Phase and the Post Processing Phase. The Pre-Processing Phase comes before the training phase and generally involves getting the input ready for learning by adding further information to the text in the corpus. This usually involves identifying the important term in the text and using Natural Language Processing techniques to find the linguistic properties of the text thus making it easier to identify the relevant data. Some common shallow NLP components

used for this purpose are Tokenization, Part of Speech, Sentence Splitting [Choi, 2000] and Semantic tagging. The Post-Processing Phase involves formatting the output to accommodate the structured representation so that it can be easily processed [Siefkes and Siniakov, 2005]. Semantic annotation is then the process of finding occurrences of a particular entity or element in a larger text domain [Mika *et al.*, 2008]. Such processes are dependent on the quality of the training given to the algorithm[Mika *et al.*, 2008].

Information extraction finds its relevance in the semantic web since it aids further understanding of the text being processed. In their paper 'Linked data on the Web', Bizer *et al* explain linked data as the employment of *"RDF and HTTP to publish structured data on the web"* [Bizer *et al.*, 2008]. They go on by emphasising that the relevance of data stands in the connection between sources which hold data to make it more meaningful [Bizer *et al.*, 2008]. The term *'Linked Data'* was coined once again by Sir Tim Berners-Lee in 2006 [Berners-Lee, 2009]. In his publication, Berners-Lee made an effort in explaining that the semantic web is different from the other web because it is not just about putting data on a source. He emphasised that linking data is important so that *"a person or a machine can explore the web of data"* [Berners-Lee, 2009]. It is exactly at this point where one starts to appreciate the value of information extraction as a method of finding important parts of the document to be able to relate to others, relevantly.

## 3 Information Extraction Algorithms

Algorithms which involve learning are normally categorised into two main classes: *Supervised* and *Unsupervised*. In supervised learning, human intervention occurs before the algorithm starts learning (example in IE: annotating the training corpus going to be used as input) or at particular stages in the learning stages (example in IE: modifying the rules developed by the algorithm). For unsupervised learning, human intervention is kept to a minimum and usually only involves selecting the training corpus (annotation and rule modifying is done automatically by the algorithm) [Mitchell, 1997].

Another characteristic of Extraction algorithms is the way in which algorithms learn after inducing a rule or a pattern. The approach can be either *Bottom Up* or *Top Down*.In a Top-Down approach learning basically starts from scratch by getting a generic rule or pattern and develops it into a new rule. The generic rule or pattern has a high recall but a low precision. The precision will eventually increase through the algorithms customization of the rule. When the precision increases to its maximum, the algorithm will stop modifying the rule. In a Bottom-Up approach learning starts with a fully customized rule or pattern (i.e. it contains all features available) and reduces it to a more generic one by dropping some of its features. This will generally start with a rule or pattern high in precision but low in recall and will know it has perfected the rule when a maximum threshold error value defined by the user is reached [Kushmerick and Thomas, 2003].

Information Extraction Systems can be used to create database records with concepts learned from the documents [Chieu and Ng, 2002]. These concepts are normally referred to as slots. *Single Slot* and *Multi Slot* extraction refer to the number of concepts learnt simultaneously in each document. Single-Slot extraction means that, for each document, only one concept can be learnt at a time. On the other hand, in Multi-Slot extraction, each document can contain several concepts [Chieu and Ng, 2002] and the system can also extract relationships between those concepts.

There are three main types of Information Extraction algorithms [Siefkes and Siniakov, 2005] which are namely *Rule Learning*, *Knowledge Based* and *Statistical Approaches*. We will focus on the former type and below follows a detailed illustration of the major sub-classes in which Rule Learning Algorithms are organised:

**Covering Algorithms** The algorithms in this category adopt a special type of inductive learning that is based on divide and conquer. They rely on a predefined target structure and require a fully annotated training corpus where all the relevant information is labelled [Siefkes and Siniakov, 2005]. The algorithms then induce rules based on the annotated training corpus that extracts the slot fillers that represent the relevant information in the text. The instances that are mapped by the learned rules are removed from the corpus and the algorithms continue to learn rules for the remaining instances in the corpus until every instance is covered by a rule or according to a predefined setting [Siefkes and Siniakov, 2005]. An example of this algorithm is the $LP^2$ which is described in section 3.1.

**Wrapper Induction** Muslea *et al* [Muslea *et al.*, 2003] refer to Wrapper Induction (WI) as algorithms aiming to learn extraction patterns, called *wrappers*, by extracting relevant information usually from collections of semi-structured or structured documents that share the same domain specific information. At execution, wrappers are used on unseen collections of documents (which share the same domain specific information used in training) to fill predefined data structures with the information extracted [Muslea *et al.*, 2003]. Two prominent algorithms which are placed in this class are the STALKER algorithm [Sigletos *et al.*, 2004] and the Boosted Wrapper Induction (BWI) which is described in more detail in section 3.2.

**Pattern and Template Creation** This category reduces the amount of human effort and knowledge needed for preprocessing the corpus that is usually adopted in other rule-based learners to generate the extraction rules. Syntactic and lexical resources are provided to cover word semantics in the domain in order to compensate the lack of human interaction. Usually the induced patterns have a simple syntactic structure and the final patterns are chosen using statistical approaches from a large amount of initial patterns[Siefkes and Siniakov, 2005].

**Relational Learners** This group of learners is very similar to the covering algorithms category in the fact that they remove the instances that are covered by the rules induced and continue to work with the remaining instances in the corpus. However in this category, the algorithms consider relations between unlimited combinations of

features instead of limiting to a predefined one [Siefkes and Siniakov, 2005]. An example of a relational learner is the SRV learner.

### 3.1 LP$^2$

Ciravegna [Ciravegna, 2001] introduces LP$^2$ as *"an algorithm able to perform implicit event recognition on documents of different types, including free, structured and mixed ones."*. This type of covering algorithm borrows some ideas from Wrapper Induction and avoids data sparseness in natural language texts by making use of shallow NLP to add semantic meaning, while keeping the same effectiveness on semi-structured and structured texts. This algorithm splits the corpus into the *Training Corpus*, which will be used in the learning stage to induce the rules, and the *Testing Corpus*, which will be used to evaluate the learned rules [Ciravegna, 2001]. LP$^2$ performs Adaptive Information Extraction because it is capable of being ported to new scenarios without the need of an IE expert to configure it. Also it outperforms other learning algorithms and gives the best results in a large number of test cases made up of from different domains [Ciravegna, 2001]. It was for these reason that this algorithm has been chosen to be a base algorithm, together with BWI, for the purpose of this project.

Before inducing rules, LP$^2$ uses external linguistic tools over the corpus to give syntactic and semantic meaning to the text. A linguistic pre-processor is used to perform tokenization, morphological analysis, part of speech tagging, gazetteer lookup and generic dictionary lookup on the text corpus. This algorithm then uses the annotated text to generate rules for extracting the relevant information which is labelled with SGML tags by the user.

### 3.2 BWI

Freitag and Kushmerick [Freitag and Kushmerick, 2000] claim BWI to be a system *"that performs information extraction in both traditional (natural text) and wrapper (machine-generated or rigidly-structured text) domains"*. This algorithm learns simple extracting procedures, called wrappers, while applying boosting (a machine-learning technique) to improve the performance of simpler algorithms. The basic theory behind this technique is that *"finding many rough rules of thumb can be a lot easier than finding a single, highly accurate prediction rule"* [Schapire, 2003]. Generally wrapper induction algorithms try to generate one accurate rule that optimally has high precision and recall. In BWI, many low precision rules will be induced that individually are considered as weak but collectively make-up an accurate classifier. AdaBoost, the boosting algorithm, is applied by Freitag and Kushmerick in Wrapper Induction, to induce a number of weak learners (having a confidence value) that will form the ultimate wrapper.

## 4 Methodology

In this section we will show how we combined the algorithms illustrated in sections 3.1 and 3.2 to create RULIE. RULIE exploits the main features of the LP$^2$ and the BWI algorithms to produce a set of extraction rules which contain most of the advantages of both algorithms.

This system uses the Carnegie Mellon University (CMU) Seminar Announcements dataset as corpus for both training and testing. The CMU Seminar Announcements is a dataset 485 e-mails labelled by Freitag [Freitag, 1998] which contain information on seminars that were held at the CMU. This is a classic corpus used to evaluate Information Extraction algorithms.

### 4.1 Design

RULIE is divided into two parts as illustrated by Siefkes and Siniakov [Siefkes and Siniakov, 2005], i.e. the **Training Function** which represents the Learning Phase where extraction rules are learned from the training corpus and the **Testing Function** which represents the information extraction phase where the extraction rules learned in the first part are used to extract information from the Test Corpus.

**Training Function**

This process starts by first taking the corpus containing the examples and adding further annotations. Using GATE [Cunningham, 2002], it assigns syntactical and semantic information to each term in the document. The resultant *Annotated Corpus* is a representation of the training corpus but with added semantics. Subsequently the rule learning procedures start by taking the annotated corpus and generating rules which are too specific and considered to be weak. Later, the *Weak Rules* are generalised by reducing their length and relaxing their constraints in order to make them more generic thus increase recall and precision. Obviously, some rules will be discarded since this process would render them useless. The *Generalised Rules* are then tested by mapping them to documents in the training corpus. These *Mapped Rules* are then unified together by using the AdaBoost algorithm. After these rules are tested by calculating their recall and precision, they are sorted according to their recall and precision rating. This function then returns the final set of Extraction Rules.

**Testing Function**

In this function, a process similar to the training function is followed. An unseen corpus without annotated examples is used. This is then annotated using GATE as illustrated in section 4.1. By testing the annotated corpus with the *Extraction Rules* which were learned in the Training Function, RULIE extracts the relevant information present in corpus. The *Extracted Information* is then compared and statistical data on the effectiveness of the rules together with the overall performance of the system are computed. This function then returns these *Quantitative Metrics* together with the information extracted.

### 4.2 The Algorithm

The RULIE algorithm as illustrated in Section 4.1 is designed to induce a rule for every tag present in the document. Like the LP$^2$ and the BWI algorithms, RULIE will induce rules for the opening tags and the closing tags separately and thus it would learn a rule for an opening tag and then it would learn another rule for its closing tag. A rule is composed of two sets. A set representing the words found before the tag and another represent the words after. The size of the window of words can be set by the user during the training phase.

For each word **w**, the system will also consider its semantic category, its lexical category, the case of its first letter and its lemma. The position of the tag will also be stored for future reference. This rule representation is more similar to the of the $LP^2$ algorithm rather than the BWI approach. The $LP^2$ can give more significant and expressive wild cards than the BWI since it uses an external linguistic pre-processor. RULIE also uses an external pre-processor to develop additional semantic wildcards for the words in the rule.

Figure 1 shows the flowchart of the RULIE algorithm. The flow of the algorithm shows that it is based on two main loops where in the first loop, rules are initialised as shown by algorithm 4.1 and subsequently generalised as shown in algorithm 4.2. In the second loop, the mapped rules are revisited and the unification takes place as explained below.

In the function 'CreateInitialRule' illustrated in algorithm 4.1, a tag is found in the instance and stored in 'TagLocation'. Rule **R** is then built by individually storing the words before the tag and the words after the tag. Finally, the Rule is returned by the function.

**Algorithm 4.1:** CREATEINITIALRULE($Instance$)

$TagLocation \leftarrow position(Tag, Instance)$
$R.WordsBefore \leftarrow wordsbeforeTagLocation$
$R.WordsAfter \leftarrow wordsafterTagLocation$
**return** $(R)$

The generalisation of the rules illustrated in algorithm 4.2 occurs by going through all the constrains in a particular rule **R**. Each constrain is removed from the rule one-by-one in order to relax the rule and each time a constraint is removed, a new rule is created. The resultant rules are finally returned by the function.

**Algorithm 4.2:** GENERALISATIONS($R$)

**for each** $Constrain : C \in R$
**do** $GeneralisedRule : G \leftarrow R - C$
**return** $(G)$

The last key element of the RULIE algorithm is the Unification module. Similarly to the BWI, this module employs the AdaBoost algorithm to combine the mapped rules in a single rule which is finally returned by the module.

## 5 Experimental Results

In Section 4 we showed how the CMU Seminar Announcements [Freitag, 1998] were used as a corpus to test this system and how these announcements were going to be divided into a training corpus and a test corpus. This division is obtained using a N-Fold Cross Validation approach, an evaluation technique that randomly partitions the data into N sets of equal size and each time the learning algorithm is run one of the N sets will be the test corpus while the remaining N-1 sets will be the training corpus [Zhu and Rohwer, 1996].

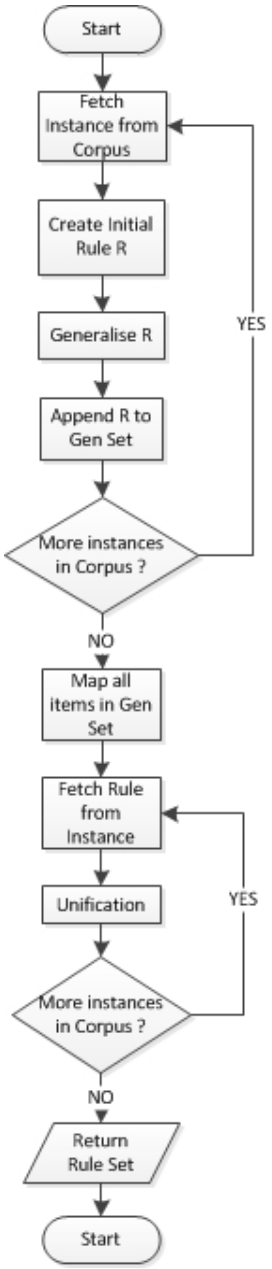


Figure 1: Flowchart of RULIE Algorithm (Source: Authors)

For the purpose of this project we are going to use 2-Fold, 5-Fold and 10-Fold Cross Validation to ensure that the system is properly evaluated on different division of the corpus and so that we can compare the results from one fold with the results of another. This ensures that no announcement used in the training phase would be used in the testing phase and vice versa. The system basically evaluates the four main fields in the CMU Seminar Announcements: The **Speaker** that is going to talk at the seminar; the **Location** where the seminar is going to be held; the **Start Time** and **End Time** of the seminar.

In our evaluation of RULIE, we compared the perfor-

| | Precision | Recall | F-Measure |
|---|---|---|---|
| **RULIE** | 0.895 | 0.908 | 0.898 |
| $LP^2$ | 0.918 | 0.830 | 0.865 |
| BWI | 0.893 | 0.803 | 0.838 |
| RAPIER | 0.905 | 0.725 | 0.790 |
| SRV | 0.733 | 0.795 | 0.758 |
| WHISK | 0.765 | 0.633 | 0.655 |

Table 1: Comparison of IE Algorithms

mance measures of this algorithm with those of other prominent information extraction algorithms namely: $LP^2$, BWI, RAPIER, SRV and WHISK. The CMU Seminar Announcement was used since it can be considered as being a Gold Standard to test the performance measures of the above mentioned algorithms.

Table 1 presents a comparison of the Precision, Recall and F-Measure of all these algorithms including RULIE. The resuls in table 1 show that RULIE performs significantly better than the other algorithms used in information extraction. These results are pictorially displayed in figure 2.

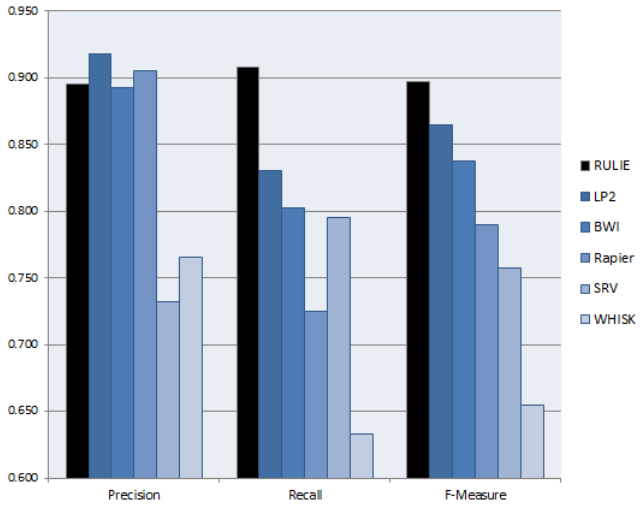


Figure 2: Performance measure of RULIE against other IE algorithms (Source: Authors)

## 6 Future Improvements

The efforts till now in RULIE were naturally revolving around its design, development and devising a system to evaluate it. More effort will be dedicated to evaluate the RULIE algorithm on different datasets.

The semantic web is the ultimate goal for this algorithm. A system implementing RULIE can be easily transformed into a web service and thus making it portable and accessible by both humans and other agents on the web. By creating an online version, the system can be used to produce liked-data [Berners-Lee, 2009].

## 7 Conclusion

In this paper we explored the importance of Information Extraction algorithms. This was done to develop RULIE our information extraction algorithm and to subsequent create comparisons with other Information Extraction algorithms.

The key algorithms were studied and out of them $LP^2$ and the BWI algorithm were examined in more detail. We chose these two algorithms to act as models for RULIE since they share positive attributes and our research showed that they had the potential to be combined. The rule induction technique in $LP^2$ and the rule unification in BWI have been unified under RULIE in order to get more expressive types of rules that apply to more situations. It is important to outline that although RULIE was created on the $LP^2$ and BWI technique, it introduces new ideas while refining the modelling algorithms. The major change was to relate the opening and closing tag rules with one another in order to remove the need of correction rules (used in the $LP^2$ algorithm) and to make the BWI algorithm handle richer rules.

Although as discussed in section 6 there is still room for improvement in RULIE, the results obtained till now are remarkable and promising. When considering the fact that RULIE outruns algorithms which were commercially employed, one cannot fail to notice the potential of this algorithm.

## References

[Berners-Lee and Fischetti, 1999] Tim Berners-Lee and Mark Fischetti. *Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. Harper San Francisco, September 1999.

[Berners-Lee, 2001] James; Lassila Ora Berners-Lee, Tim; Hendler. The semantic web. *Scientific America*, May 2001.

[Berners-Lee, 2009] Tim Berners-Lee. Linked data. w3 Website, 06 2009. http://www.w3.org/DesignIssues/LinkedData.html.

[Bizer *et al.*, 2008] Christian Bizer, Tom Heath, Kingsley Idehen, and Tim Berners-Lee. Linked data on the web (ldow2008). In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 1265–1266, New York, NY, USA, 2008. ACM.

[Chieu and Ng, 2002] Hai Leong Chieu and Hwee Tou Ng. A maximum entropy approach to information extraction from semi-structured and free text. In *Eighteenth national conference on Artificial intelligence*, pages 786–791, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[Choi, 2000] Freddy Y. Y. Choi. Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 26–33, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[Ciravegna, 2001] Fabio Ciravegna. Adaptive information extraction from text by rule induction and generalisation. In *Proceedings of the 17th international joint conference*

*on Artificial intelligence - Volume 2*, pages 1251–1256, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[Cunningham, 2002] H. Cunningham. GATE, a General Architecture for Text Engineering. *Computers and the Humanities*, 36:223–254, 2002.

[Department and Ciravegna, 2001] Fabio Ciravegna Department and Fabio Ciravegna. an adaptive algorithm for information extraction from web-related texts. In *In Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*, 2001.

[Freitag and Kushmerick, 2000] Dayne Freitag and Nicholas Kushmerick. Boosted wrapper induction. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 577–583. AAAI Press, 2000.

[Freitag, 1998] D. Freitag. Information extraction from html: Application of a general learning approach. *Proceedings of the Fifteenth Conference on Artificial Intelligence AAAI-98*, pages 517–523, 1998.

[Kushmerick and Thomas, 2003] Nicholas Kushmerick and Bernd Thomas. Intelligent information agents. In Matthias Klusch, Sonia Bergamaschi, Pete Edwards, and Paolo Petta, editors, ., chapter Adaptive information extraction: core technologies for information agents, pages 79–103. Springer-Verlag, Berlin, Heidelberg, 2003.

[Mika *et al.*, 2008] P. Mika, M. Ciaramita, H. Zaragoza, and J. Atserias. Learning to tag and tagging to learn: A case study on wikipedia. *Intelligent Systems, IEEE*, 23(5):26 –33, septoct. 2008.

[Mitchell, 1997] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[Moens, 2006] Marie-Francine Moens. *Information Extraction: Algorithms and Prospects in a Retrieval Context (The Information Retrieval Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[Muslea *et al.*, 2003] Ion Muslea, Steven N. Minton, and Craig A. Knoblock. Active learning with strong and weak views: A case study on wrapper induction. In *In Proc. 18th Int. Joint Conference on Artificial Intelligence*, pages 415–420, 2003.

[Schapire, 2003] Robert Schapire. The boosting approach to machine learning: An overview, 2003.

[Siefkes and Siniakov, 2005] Christian Siefkes and Peter Siniakov. An overview and classification of adaptive approaches to information extraction. In *JOURNAL ON DATA SEMANTICS, IV:172212. LNCS 3730*. Springer, 2005.

[Sigletos *et al.*, 2004] Georgios Sigletos, Georgios Paliouras, Constantine D. Spyropoulos, and Michalis Hatzopoulos. Mining web sites using wrapper induction, named entities, and post-processing. In Bettina Berendt, Andreas Hotho, Dunja Mladenic, Maarten van Someren, Myra Spiliopoulou, and Gerd Stumme, editors, *Web Mining: FromWeb to SemanticWeb*, volume 3209 of *Lecture Notes in Computer Science*, pages 97–112. Springer Berlin / Heidelberg, 2004. 10.1007/978-3-540-30123-3_6.

[Téllez-Valero *et al.*, 2005] Alberto Téllez-Valero, Manuel Montes-y Gómez, and Luis Villaseñor Pineda. A Machine Learning Approach to Information Extraction. *Computational Linguistics and Intelligent Text Processing*, pages 539–547, 2005.

[Zhu and Rohwer, 1996] Huaiyu Zhu and Richard Rohwer. No free lunch for cross-validation. *Neural Comput.*, 8:1421–1426, October 1996.