Big data meets nature conservation: Automatic tools for information extraction

Ricardo Pereira
Instituto Superior Técnico
Lisbon, Portugal
ricardo.r.pereira@tecnico.ulisboa.pt

ABSTRACT

Biodiversity has been declining globally while the scientific community has been thinking and developing models to understand and stop this decline. Over the years, the number of scientific articles with data collected by scientists has been increasing and, due to the dispersion of information, it has become almost impossible to gather all the data of an high-level taxonomic group.

In this work we present a system capable of answering the following questions: "Is it possible to build a tool capable of extracting the information available in scientific articles and selecting the one that may correspond to the selected physiological characteristics?" and "Is it possible to take advantage of the user's knowledge to improve the effectiveness of this tool?". The system receives as scientific articles, extracts data, about physiological characteristics of the species being studied, from those articles and classifies it, using regular expressions and machine learning techniques.

Author Keywords

Biology; Machine Learning; Regular Expressions;
Information Extraction

INTRODUCTION

Life is what sets us apart from all the other planets we know and diversity plays a key role in maintaining this [15]. However, biodiversity has been affected by several kinds of problems and has been suffering a serious decline. These problems, with human and natural origins, have led to the extinction of many species. This is the phenomenon that makes it so important to collect information about living beings and the development of models that can guarantee their protection [23].

To solve this issue, the scientists have done research and saved the result of their work through the publication of scientific articles. But the number of articles has been increasing exponentially over the years [19]. With the creation of computers and the development of computer science, problems, such as obtaining all the transversal information about an animal species or even an entire taxonomic group can be solved, since all this information can be stored in structures such as a database, for example. And later be available just a click away.

There are already some tools that use information extraction techniques to extract data from documents and web sites and running a brief Internet search we can find some of them, such as: GATE [11] (General Architecture for Text Engineering), which includes an information extraction system called ANNIE (A Nearly-New Information Extraction System), or polyglot[1].

In the following document, we explain the elaboration of a tool capable of using existing models to extract data from scientific articles, present such data to the user and receive feedback from the user in what concerns the accuracy of the data extracted. In addition, the tool is complemented, with a machine learning algorithm that uses the feedback provided by users to improve the classification process. The extraction model is based in natural language processing techniques and regular expressions.

BACKGROUND AND RELATED WORK

Information Extraction is what we call the process of extracting structured information such as entities, relationships between entities, and attributes describing entities from unstructured sources [21]. Based on the Natural Language Processing (NLP) community this is now a topic that involves many more disciplines like machine learning, information retrieval, database, Web, and document analysis. Consequently, it has undergone a great evolution over the years, since the first tasks were based only on the identification of named entities, while nowadays it contemplates a set of new tasks such as establishing relationships between these entities [21]. By increasing the number of possible applications for this technique, it is now possible to apply it to biology or science in general.

According to [18], most Information Extraction Systems have components in common. They are called domain-independent components and usually consist of components that aim to carry out the linguistic analysis, with the following steps:

- **Meta-data analysis -** extraction of elements like the title, the body and its structure and the document's date.
- **Tokenization -** text segmentation in tokens and their respective type classification.
- **Morphological analysis -** extraction of the morphological information of each of the tokens.
- **Sentence/Utterance boundary detection -** segmentation of text into sentences (sequence of lexical items together with their features).

[1] http://polyglot.readthedocs.io/en/latest/

- **Named-entity extraction -** detection of named entities like organizations, currencies, geographical references, etc.
- **Phrase recognition -** Recognition of local structures such as noun phrases, acronyms, abbreviations, etc.
- **Syntactic analysis -** Analysis of the syntactic function of each token that is part of a sentence. It can be deep, i.e., all possible interpretations and grammatical relations within the sentence, or shallow, i.e., the analysis is made only to non-recursive linguistic structures and phenomena like ambiguities.

All these steps, as well as the two components pointed by the authors, are represented on Figure 1, being part of a typical architecture of an information extraction system.

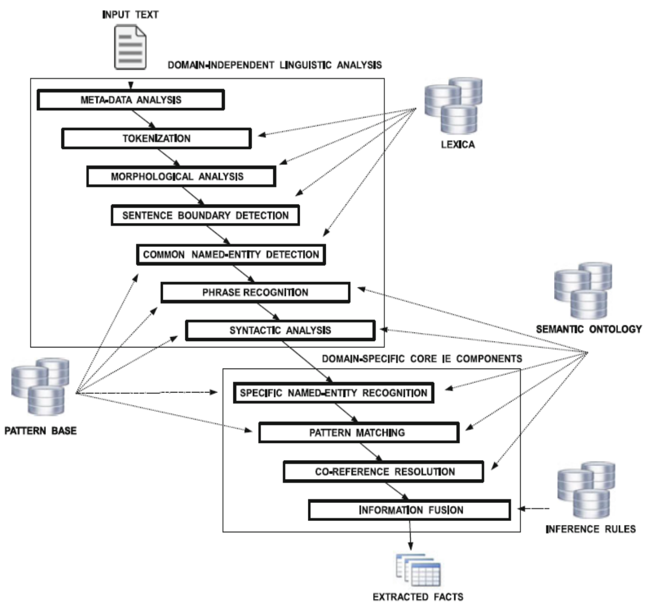

Figure 1. Typical architecture of an information extraction system [18].

Related Work

Several information extraction systems have been created over the years, aiming to collect data related to biology from scientific articles.

Despite having, usually, a well defined global structure (title, authors, keywords, references, etc.) and few variances from document to document, the main problem in its processing is the form of the content, which can either be composed only by text or can also contain images or tables. There are other problems related to the subject itself and the documents that deal with subjects related to biology are no exception [21].

Therefore, some characteristics, as the constant update of the language used due to new discoveries [12] or the linking of structures by words like *and* or *or* [4], have already been recognized as obstacles to the perfect processing of biology related documents.

There are several strategies for extracting information, namely approaches using dictionaries, rules, using machine learning algorithms and hybrid strategies, which merge two or more approaches into a single system.

Next up we will describe some existing systems, according to the strategy used for the extraction of information present in scientific articles within the scope of biology.

Dictionary-Based

Dictionary-Based strategies consist in having one or more lists composed by terms that are matched against a text and the result is the list of terms that appear in both. The advantage over rule-based systems is that the dictionary can list references to other knowledge sources. Nevertheless, it is not easy to make a list with all the necessary terms when the knowledge base is too large and when this knowledge base is constantly updated [14].

Information extraction systems, developed, within the scope of biology, that use dictionary-based algorithms essentially serve to recognize named entities (NER) or taxonomic names (TNR) [17].

The TaxonFinder[2] system is a TNR tool, that uses this type of algorithm, which recognizes taxonomic names in documents, comparing all the words in this document with several word lists from a version of NameBank[3].

The system splits the document into words and computes a comparison between them and the words in the lists. Whenever the system finds a capitalized word, it checks if the word is in the genus or in the above-genus list. If it is in the genus list, it can be returned as a name. If it is in the genus list, the system will check the "species-or-below" name list. If it is in that list, the next words are analyzed, until a complete polynomial is returned. If the next word is not in the list, the name is returned as a genus [1].

Of course the major disadvantage of this system is that the dictionary has to be constantly updated, running the risk of not being able to find new names, although it can can discover new combinations of known names.

Rules-Based

Rule-based methods usually work by establishing a set of rules either manually or through automatic learning, and apply them to a text [1]. A rule consists of a pattern and an action. This pattern is defined, for example, by a regular expression. When a pattern matches a sequence of tokens of the text under consideration the consequent action is performed [2].

The manual creation of the rules requires the participation of specialized personnel and requires much effort. Automatic methods contemplate two approaches: top-down and bottom-up. The top-down approach requires the rules to be defined first so that they can be applied to the maximum number of training instances and system will learn and specify more rules "by taking the intersections of the more general rules". On the other hand, the bottom-up approach rules are defined based on training instances and then generalized [2].

[2] http://taxonfinder.org/
[3] http://www.ubio.org/index.php?pagename=namebank

The main disadvantages of this strategy is that building the rule set manually may take a large amount of work and the set only applies to a given domain. In addition, it's practically impossible to create a completely effective rule set. On the other hand, the strength of using this strategy is the fact that, comparing with a dictionary-based strategy, it is capable of handling variations in the word's order or in the sentences structure [1].

There are few information extraction systems that rely solely, and exclusively, on rules to perform their task [9]. Instead, this approach is more often used to improve the results of other tools by combining it with other methods of extracting information. An example of this is the FAT (Find All Taxon names) system, which uses another system,TaxonGrab [6], and tries to improve it using rules [8]. Figure 2 represents the classification process of this system.

According to [8], the idea of their approach is to pick up the parts of the text that are already classified, as taxonomic names (precision rules), and as not being taxonomic names (recall rules), and use those already classified parts to build lexica and statistics that will be used to classify the rest of the text.

In a first pass, the system uses the parts already classified to detect all sequences of words equivalent to that rule. The second step is to do the same but using the recall rules. Then, the results of the last steps is used to build another lexica that will be applied to the text that did not match in any of the first two steps. The words, or phrases, that remain with uncertain classification are then subject to analysis by a word-level language recognizer to be classified.

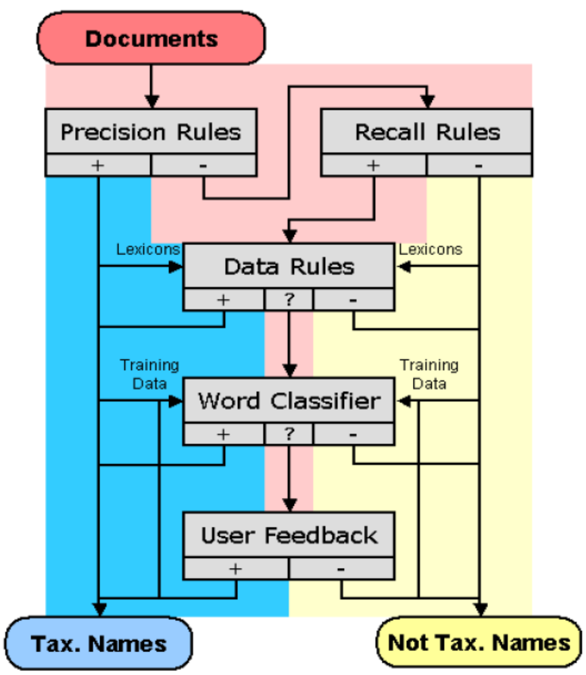

Figure 2. FAT Classification Process [8].

The system Protein Active Site Template Acquisition (PASTA) [20] is another example of a system that uses this type of approach to extract information, in this case, about the roles of amino acid residues in protein molecules.

This system exploits basic templates to extract information from the text. These templates store information about an *entity*, a *relation* between two entities or a *scenario* and contain one or more slots of information [20].

Then the system executes five major tasks: text preprocessing, terminological processing, syntactic and semantic analysis, discourse interpretation, and template extraction [20].

Machine-Learning-Based

This type of strategy aims to create an algorithm that learns to extract information more effectively than the previous strategies. The algorithm may (supervised learning) or may not (unsupervised learning) receive a training set. There are also tools that combine these two approaches [22]. The learning is done through the establishment of rules using statistical procedures [1]. The first approach manifests a major disadvantage because collecting a quality training set can be hard to do, especially in some areas, such as biology, for the reasons mentioned above [22].

The inclusion of machine learning algorithms, in this type of system, is more recent than the other approaches and only since the mid-1990s this technique became dominant against the others [16].

The system developed by Cui, Boufford and Selden is an example of the use of unsupervised machine learning for semantic annotation inside the biology scope, in this case, the annotation of morphological descriptions of whole organisms [16].

These types of systems are very closely related to information extraction systems, since they are both based on the discovery of the semantic role that a word plays in a given text.

The developed system makes use of a Bootstrapping-Based Unsupervised Machine Learning Algorithm (see Figure 3, which is a process that begins with a small group of known items that are used iteratively to know new items. The only external resource used is WordNet, that is an online lexical reference system [7].

The first steps of the algorithm include normalizing the text and segmenting the documents into clauses. The normalization involves the conversion of the uppercase letters to lowercase and the standardization of the use of hyphens.

The next steps are a preparation for the core bootstrapping modules, in which sets of tokens (like stop words) and known words are loaded, as well as their semantic roles. The module also learns nouns, in their plural and singular forms, and annotates clauses with distinct patterns.

In the core bootstrapping modules, inferences are made between subjects and boundary words. The remaining unknown words are tagged according to conventions determined by experience in this type of tasks. Then the subject is used to annotate each clause.

The secondary modules deal with more complex language features in the biosystematics literature such as: the use of an

Initiation
- Normalize text*
- Segment files into individual clauses

Pre-bootstrapping modules
- loadFiniteSets
- loadKnowledgeBase°
- learnSeedNouns*
- patternBasedAnnotation^{F}

Core bootstrapping modules
- coreBootstrapping*
- leadWordBootstrapping
- unknownWordBootstrapping

Secondary bootstrapping modules°
- adjectiveSubjectBootstrapping^{F}
- compoundSubjectBootstrapping
- wrapupBootstrapping

Post-bootstrapping modules°
- phraseClauseAnnotation
- dittoAnnotation
- pronounCharactersAnnotation
- commaAsAndAnnotation
- refineModifiers^{T}
- normalizeAnnotations

Figure 3. Overview of the Bootstrapping-Based Unsupervised Machine Learning Algorithm [16].

adjective for a noun as a subject or conjunctions like "and" or "or" to form a compound subject. Finally, using the new knowledge to annotate other clauses.

The knowledge base is constantly updated with the information obtained during execution allowing the algorithm to learn by itself. After completing these three steps the algorithm should have learned enough to annotate directly the rest of the clauses, and this is what it will do during the post-bootstrapping modules.

NetiNeti [13] is another system that uses machine learning during the information extraction. In this case, is used a supervised machine learning algorithm that involves probabilistic classifiers, Nave Bayes and Maximum Entropy, to estimate the probability of a label given a word and its context.

Hybrid Systems

Lately, in order to improve the effectiveness of the information extraction systems, it was concluded that the path to be taken was to mix the various approaches that exist, so that their strengths could complement each other. And now this hybrid approach is the most used in the creation of this type of systems [21].

BioRat, developed by Corney, Buxton, Langdon and Jones [5], is an example of one of these systems, since it uses both dictionary-based and rules-based approaches to perform biomedical information extraction.

Being based on the GATE toolbox [11], this system uses it to label words according to their parts of speech but with small changes on the gazetteers and templates.

Gazetteers are lists of words (dictionaries) used for the Name Entity Recognition. BioRat uses gazetteers from three different sources: MeSH[4], Swiss-Prot[5] and hand-made lists. Being the hand-made gazetteers created with the help of domain experts.

The templates are a representation of a pattern, matched by the text, that allows the system to extract information automatically. In short, they are predefined slots that the system tries to match with the text.

Another example of one of these systems is Caramba [10]. This system is divided into three tasks: Concept Extraction, Assertion Annotation and Relation Annotation.

The first task is based on a machine-learning method that depends on a linguistic analysis, whose output is represented in the form of n-gram tokens, typographic clues and semantic and syntactic tags for each token. Then, with the training set, a model is created using a machine learning tool called CRF++[6]. MetaMap [3] is then used to locate medical terms and their concepts and semantic types, its output is then enhanced segmenting it into noun-phrases with treetagger-chunker and searching the located terms in pre-compiled lists.

For the second task two systems were developed, one using machine learning techniques and the other using hand-made rules. The first system is a Support Vector Machine (SVM) trained with the libsvm tool[7], focusing on three types of features: contextual lexical features, trigger-based features and target concept internal features. And the second system is an extension of the NegEx [24] algorithm used to locate trigger terms indicating a negation or a probability and to determine if the concepts are within the scope of the trigger.

The final task is a classification task, in which eight relation types were considered and is used an hybrid approach based on a trained SVM and manually constructed linguistic patterns.

As it has been said this is a growing approach within the world of information extraction systems and many other systems could be described, such as TaxonGrab [6] that identifies taxonomic names using a combination of nomenclature rules and a dictionary of nontaxonomic terms.

EXTRACTION TOOL: IMPLEMENTATION

Our system follows an hybrid approach complementing the rules-based approach with an Online Machine Learning Algorithm.

The rules-based approach is present on the extraction system and is an adaptation of the work developed by Gomes (2016). This adaptation was made so that it was able to communicate with the server. In addition to the module related to the extraction of candidates, was also used the knowledge base of the same project. Table 1 contains the list of categories and words used on the extraction process.

[4]http://www.nlm.nih.gov/mesh/
[5]http://www.expasy.org/
[6]http://chasen.org/taku/software/crf++/
[7]https://www.csie.ntu.edu.tw/~cjlin/libsvm/

Category	Related Words
Body Mass	wet weight; dry weight; wet mass; dry mass; at birth; hatching; at fledging; fledgling; adult; grams; kilograms
Body Temperature	chick; adult; body; temperature; Celsius
Egg Temperature	incubation; egg; temperature; Celsius
Fledging	fledging; leaves the nest; days
Incubation	incubation; hatching; days
Total Body Water	total; body; water; content; percentage

Table 1. Categories and Related Words [9].

The process begins with the user making a folder of scientific articles available. Then, for each PDF file, the text is analyzed morphologically through the use of Natural Language Processing methods, like splitting by sentences. The forth step is to apply rules, in the form of regular expressions, to the text in order to extract only the interesting parts according to the fields defined a priori, i.e. those present in the column of categories of Table 1. Then an online learning algorithm is applied in order to improve the effectiveness of the tool, over time, depending on the input passed by the users through the response acceptance and rejection buttons.

In the interface the user can always consult the document under analysis, the fields studied, the possible answers and the phrase in which each answer is inserted, in order to be able to verify the context in which the answer is inserted and to be able to make a better decision about the response correction, the user can also write some comment that wants to see associated with the data that is to be classified. Figure 4 reflects the overall look of the tool's architecture.

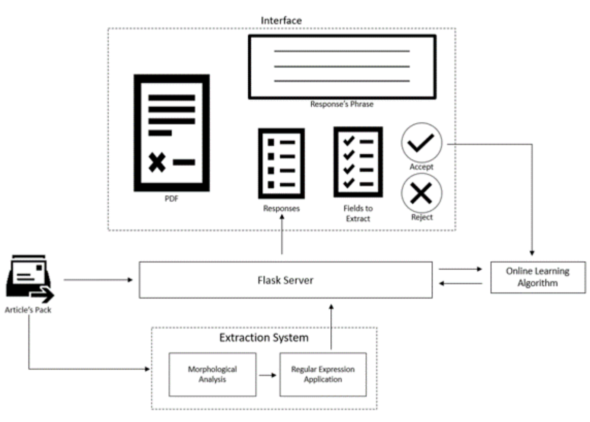

Figure 4. Extraction Tool's Architecture.

Extraction System

The extraction process consists of seven steps, three related to the morphological analysis and the other four correspond to the application of regular expressions.

The first step is to convert the PDF file text into the equivalent in a .txt file extension. To achieve this, a tool called PDFMiner.six[8] was used, which makes it possible to manipulate the data present in the article in order to obtain the expected results.

Then, the text is divided by sentences using the Natural Language Toolkit[9] (NLTK) library, being given due treatment to situations where there are words separated by hyphens or line changes.

The next step is to use the StanfordPOSTTagger[10] library to go through all the sentences in the document and to search and limit it only to sentences containing numbers.

After filtering the relevant sentences of the document, the process enters the regular expressions phase. At this stage, regular expressions are used to find phrases that contain words referring to the metrics previously identified (grams, celsius, etc.) and the words related to each category (see Table 1).

Having found any phrase in the previous step, the values are extracted and normalized. Values that are written in their extensive form are converted to digits using the word2number[11] module. The normalization is done as in the project developed by Gomes (2016).

At the end of this process a list is given to the server that contains, for each identified phrase, the values, the context in which these values are inserted and the class to which they belong, being classified by default as belonging to the class OTHER but, later, this by the class assigned by the machine learning algorithm (see Figure 5).

The result of this process is cached using a module called pickle[12] that allows us to serialize and de-serialize data structures in order to use them later, making it unnecessary to repeat the procedure for the document in question.

Learning Algorithm

The classification system, which includes the online learning algorithm, aims to assign the right class to the value that is extracted by the system. For this, the system uses some positive examples (extracted and classified manually), so that the algorithm does not begin the classification process without any knowledge, but mainly knowledge passed by the user through the interface.

The features analyzed in each value were the ones defined in the work developed by Gomes (2016):

- **Number of vocabulary words:** identification of the words contained in the phrases that are part of the previously known vocabulary (the one generated with all the training examples);
- **Number of words:** sentence size;
- **Distance between value and specific words:** recognition of the words contained in the phrases that are part of the

[8] `https://github.com/pdfminer/pdfminer.six`
[9] `https://www.nltk.org`
[10] `https://nlp.stanford.edu/software/tagger.shtml`
[11] `https://pypi.org/project/word2number/`
[12] `https://docs.python.org/3/library/pickle.html`

Phrase	Value	Context	Class
On 27 May the nest had two chicks 1 and 3 days old, a natural egg which subsequently hatched, and the radioegg; the radioegg and antenna were removed from the nest.	1	1 and 3 days	OTHER
	3	1 and 3 days	OTHER
We have excluded the final 3 days because of the presence of the hatched chicks, and there were about 64 hours for which there is no record because at some egg positions the radio signal was not picked up by the antenna.	3	3 days	OTHER

Figure 5. Example of output passed to the server.

previously known vocabulary and calculation of the distance between them and the values;

- **Interval verification with parameterized numbers:** knowing the mean and median values of the various categories, we checked if the analyzed value was close to those values, returning 1 if it is and 0 if it's not;
- **Distance of parameterized numbers:** numerical distances of the value under analysis compared to the mean and median values.

Initially, the algorithm is trained with about 70 positive cases, collected and classified manually, and is fitted to the vocabulary resulting from the phrases extracted from the set of documents uploaded by the user. Once the algorithm is created, it will be saved and will only be deleted if the user decides.

With that being done, the algorithm is only called in two occasions: every time the process of extraction of a PDF is finished, to classify the extracted values, and when the user clicks the 'next' button,to be trained with the changes made and also to reclassify the remaining values of the document being analyzed, causing the user to see data always updated according to the latest version of the algorithm. All data changed by the user is added to the file containing the positive cases, making possible to use them to train a new algorithm. Figure 6 represents the whole process described previously.

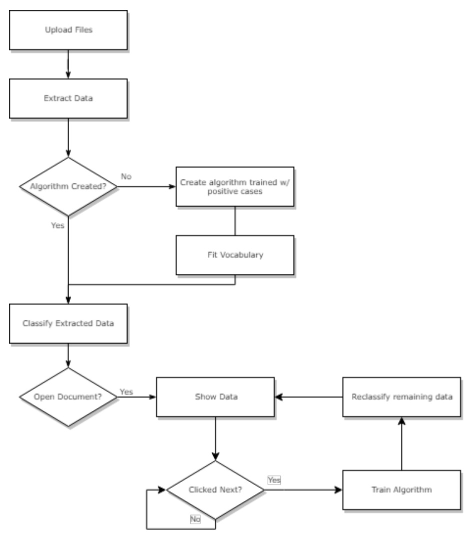

Figure 6. Flow chart with an overview of the whole process.

Interface

The interface is what allows the system to obtain the user's knowledge and consequently to develop its classification ability. This interface consists of two main pages and was tested on the Firefox 62.0 and on the Opera 57.0.3072.0. The first page is the homepage, where the user can upload and see which files are already available for review. The second is the page that shows the data extracted from each document, in which the user can see the data, the sentences in which they are inserted, the entire document and assign a class to the same data. Figures 7 and 8 are screen shots of each of these pages and the description of the functionality of each of its elements.

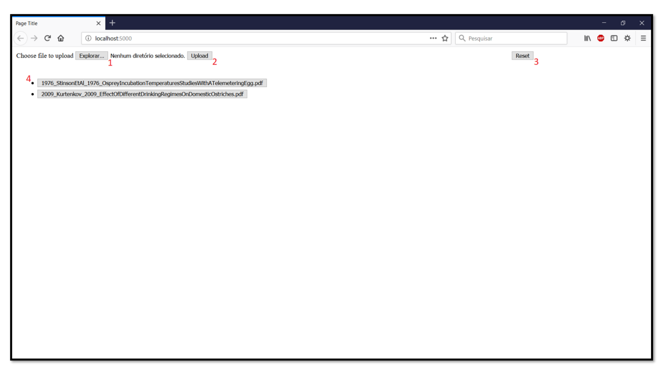

Figure 7. Interface Homepage.

1. Button to find the folder with the documents to analyze.
2. Button to upload the folder.
3. Button to reset the classifier and the vocabulary.
4. List of documents ready to analysis.

TESTS AND RESULTS

After the implementation were made some tests that were useful to make some decisions about the system, such as

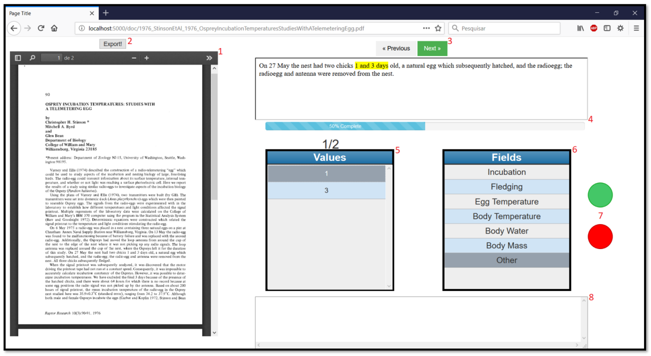

Figure 8. Interface Document's Page.

1. Document's PDF (with features like search or go to).
2. Export button to export the data to a .csv file.
3. Next and Previous buttons to change the sentence under analysis.
4. Sentence in which the data appears.
5. Values extracted from the sentence.
6. Possible classes for these values.
7. Acceptance and Rejection button (green to confirm that the class is correct and red to reject the value as a relevant data)
8. Comments box.

choosing the learning algorithm. Was also made, a prediction of the learning curve of the algorithm and an analysis of the results obtained by the system.

Evaluation Methodology

The execution of these tests involved the use of about 50 articles that, after the extraction process, resulted in more than 500 values. The algorithms used were recommended by the scikit-learn [13] library as being suitable for the incremental learning that we wanted to carry out, this is due to the fact that they have implemented a partial fit function, being able to learn from a mini-batch of instances.

Class	Group Size
Body Mass	12
Body Temperature	22
Egg Temperature	30
Fledging	20
Incubation	24
Total Body Water	12
Other	409
Total	529

Table 2. Dataset size by class.

As we can see on Table 2 the group of values classified as part of the "OTHER" class was too large, we decided that the best way to test it was with different sizes. First we tested without changing the data resultant from the extraction process, then with the group of the class "OTHER" having the same size as the sum of the other groups and then without any data classified as belonging to this group.

Effectiveness Assessment

The first tests aimed to collect information about the performance of each of the algorithms with respect to the classification of all extracted data.

We followed the Leave One Out approach, with the help of the LeaveOneOut [14] library from sckit-learn, testing in each iteration the classification of a value using the rest as training values until all the values were classified. Then, we calculated the averages for accuracy, precision, f1-score and recall of each of the algorithms.

As we can see on Figure 9, the worst results come from the algorithm Passive Aggressive that in none of the test variants demonstrates acceptable results, both the Perceptron, the SGD and the MLP seem to lose effectiveness as the distribution of test data classes becomes more uniform, yet the behavior of these three classifiers may be due to the type of features, given that the data provided is mostly discrete and some features are even binary. This type of data seems to benefit the operation of Bernoulli Naïve Bayes and Multinomial Naïve Bayes algorithms, and especially the former seems to have a very good overall performance that is improving as the number of OTHERs is decreasing. This behavior does not occur in any other algorithm, which may indicate that the other algorithms are almost always classifying the data in the same way in the first two variants of the test, which proves the usefulness of this test.

Learning Rate Assessment

We also measure the learning curve of each of the algorithms, to see if there were any who learned notoriously faster than the others, in order to minimize the time the tool has a low success rate, when being used for the first time.

First we tested the evolution at each iteration of the leave-one-out approach, then was defined that the first third of the values extracted would work as training values and the rest as test values. Starting from a training set with only one value and adding a new value to this set at each iteration, also making a new prediction at each iteration. To avoid getting results based on luck or chance, the results demonstrated were obtained after performing this test as many times as the size of the training set, and then averaged. The training set was randomly defined using the train_test_split [15] library of scikit-learn.

The results, shown below, only correspond to the two algorithms with better results in the previous tests, Bernoulli Naïve Bayes and Multinomial Naïve Bayes, and for the last variant of the test because we believe is the one closer to the reality.

[13] `http://scikit-learn.org/stable/modules/scaling_strategies.html`

[14] `http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.LeaveOneOut.html`

[15] `http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html`

		Average 1	Average 2	Average 3
Perceptron	Precision	0,60	0,51	0,04
	Recall	0,77	0,10	0,20
	F1-Score	0,67	0,17	0,07
	Accuracy	0,77	0,09	0,20
Bernoulli NB	Precision	0,71	0,68	0,82
	Recall	0,79	0,70	0,73
	F1-Score	0,75	0,69	0,77
	Accuracy	0,79	0,69	0,73
SGD Classifier	Precision	0,61	0,31	0,15
	Recall	0,60	0,33	0,14
	F1-Score	0,60	0,32	0,14
	Accuracy	0,60	0,33	0,14
Passive Aggressive	Precision	0,00	0,25	0,06
	Recall	0,05	0,50	0,25
	F1-Score	0,00	0,33	0,10
	Accuracy	0,05	0,50	0,25
Multinomial NB	Precision	0,62	0,38	0,64
	Recall	0,18	0,32	0,62
	F1-Score	0,28	0,35	0,63
	Accuracy	0,18	0,32	0,62
MLP Classifier	Precision	0,61	0,31	0,23
	Recall	0,63	0,28	0,21
	F1-Score	0,62	0,29	0,22
	Accuracy	0,63	0,28	0,21

Figure 9. Effectiveness Assessment. Average 1 - No data change; Average 2 - Number of OTHERs equals the sum of the remaining; Average 3 - No data classified as OTHER

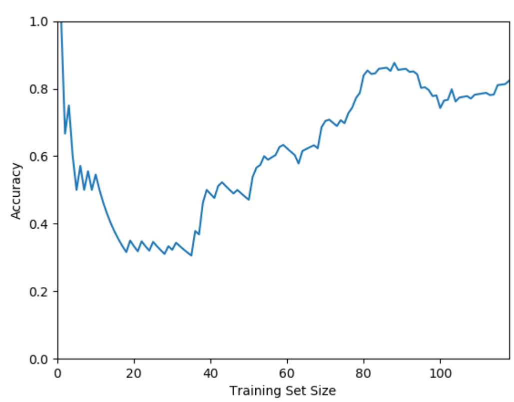

Figure 10. Bernoulli learning curve following the leave-one-out approach.

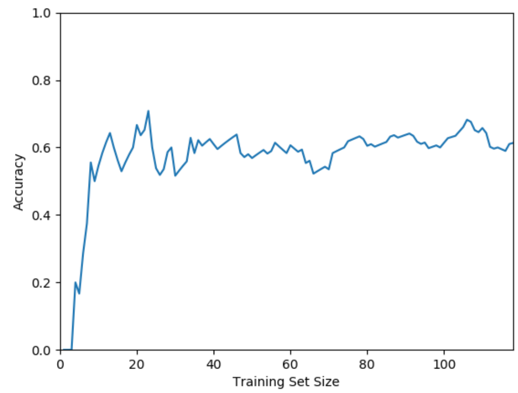

Figure 11. Multinomial learning curve following the leave-one-out approach.

On Figures 10 and 11 we can see that Bernoulli's curve shows a more or less constant growth and, in addition, a maximum point around the 80%. On the other hand, the Multinomial algorithm shows a large and fast initial growth but from then on the value remains constant, with an accuracy of about 60%.

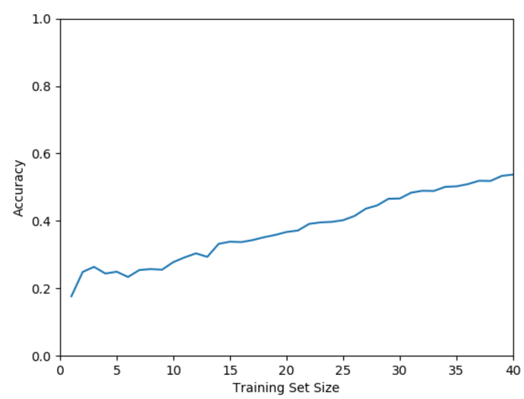

Figure 12. Bernoulli learning curve following the incremental training set approach.

As we can see on Figures 12 and 13, the Bernoulli algorithm, once again, obtains better results than the Multinomial, not only with respect to the values of the accuracy but also with respect to the evolution of this measure as the number of examples increases over the test, being the only algorithm that demonstrates a remarkably positive evolution over time.The Multinomial algorithm has a faster rise initially but then the accuracy value remains constant, at only 45%.

CONCLUSION

This work arises with the purpose of helping the scientific community to obtain information about the physiognomy of the constituent species of the taxonomic group of birds, with the main objective of answering to the following questions: "Is it possible to build a tool capable of extracting the information available in scientific articles and selecting the one

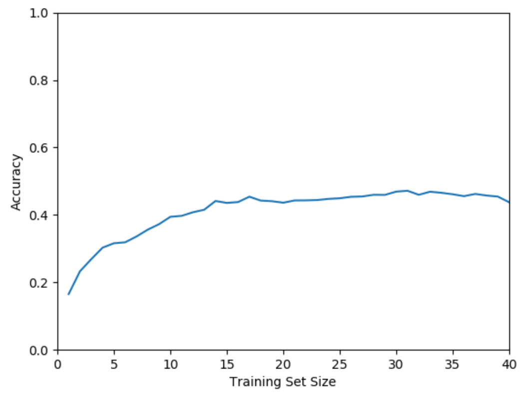

Figure 13. Multinomial learning curve following the incremental training set approach.

that may correspond to the selected physiological characteristics?" and "Is it possible to take advantage of the user's knowledge to improve the effectiveness of this tool?".

In order to answer these questions, we have built a tool capable of receiving scientific articles, extracting data that it considers relevant from the same articles, classifying the data, presenting it to the user, and taking advantage of the user's feedback to improve the classification process. The system follows an hybrid approach complementing a rules-based approach with an Online Machine Learning Algorithm.

This tool is composed of 3 modules: the extraction system, in which is made the morphological analysis and the application of regular expressions to extract data from the articles, the learning algorithm, that classifies this data, and the interface, which allows the user to evaluate the work done by the algorithm and correct it when necessary.

Through the tests made we could conclude that the Bernoulli Naïve Bayes algorithm clearly showed the best results, both in terms of absolute values and in relation to the learning rate. These results are not completely surprising because it was known in advance that this is a suitable classifier for discrete data, which is the main type of data present in the features being studied.

We also concluded that these tests were negatively influenced by the excessive presence of data classified as belonging to the class OTHER, so anyone who uses the tool should limit as much as possible the choice of this class to classify the extracted data, something that can also be corrected by increasing the number of themes covered by the tool itself.

ACKNOWLEDGMENTS

I would like thank Professor Gonalo Marques and Carlos Teixeira for their participation in various discussions, about the work developed, and the feedback given that allowed me to learn and progress over the last months.

And, also, to Professor Pável Calado, who helped me during the development of this work and guided me in the search for solutions to the challenges that have arisen until the presentation of the final version.

REFERENCES

1. A. Thessen, H. C., and Mozzherin, D. Applications of natural language processing in biodiversity science.
2. Aggarwal, C., and Zhai, C. *Mining text data*. 2013.
3. Aronson, A. R. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. *Proceedings of the AMIA Symposium* (2001), 17.
4. Cohen, K., and Hunter, L. *Artificial Intelligence Methods and Tools for Systems Biology*. Springer, 2004.
5. D. Corney, B. Buxton, W. L., and Jones, D. Biorat: Extracting biological information from full-length papers. *Bioinformatics 20* (2004), 3206–3213.
6. D. Koning, I. N. S., and Moritz, T. M. Taxongrab: extracting taxonomic names from text. *Biodiversity Informatics 2* (2005), 7982.
7. G. A. Miller, R. Beckwith, C. F. D. G., and Miller, K. J. Introduction to wordnet: An on-line lexical database. *International journal of lexicography 3*, 4 (1990), 235244.
8. G. Sautter, K. B., and Agosti, D. A combining approach to find all taxon names (fat) in legacy biosystematics literature. *Biodiversity Informatics 3* (2006), 4658.
9. Gomes, J. Extrao de informao biolgica de artigos cientficos engenharia informtica e computadores.
10. Grouin, B., and AB, A. Caramba: concept, assertion, and relation annotation using machine-learning based approaches. *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data* (2010).
11. H. Cunningham, D. Maynard, K. B., and Tablan, V. Gate: an architecture for development of robust hlt applications. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02* (2001), 168.
12. L. Hirschman, A. M., and Yeh, A. Rutabaga by any other name: Extracting biological names. *Journal of Biomedical Informatics 35* (2002), 247–259.
13. L. M. Akella, C. N. N., and Miller, H. Netineti: discovery of scientific names from text using machine learning methods. *BMC Bioinformatics 13*, 1 (2012), 211.
14. M. Krauthammer, A. Rzhetsky, P. M., and Friedman, C. Using blast for identifying gene and protein names in journal articles. *Gene 259* (2000), 245–252.
15. Marjorie L Reaka-kudla, Don E Wilson, O. E., and Henry, A. J. *Biodiversity II. Understanding and Protecting our Biological Resources*. Environment International, 1997.
16. Oakleaf, M. Writing information literacy assessment plans: A guide to best practice. *Communications in Information Literacy 3*, 2 (2009), 8090.

17. P. R. Leary, D. P. Remsen, C. N. N. D. J. P., and Sarkar, I. N. Ubiorss: tracking taxonomic literature using rss. *Bioinformatics 23*, 11 (2007), 14341436.

18. Piskorski, J., and Yangarber, R. Multi-source, multilingual information extraction and summarization. 23–50.

19. P.Larsen, and von Ins, M. The rate of growth in scientific publication and the decline in coverage provided by science citation index. *Scientometrics 84* (2010), 575–603.

20. R. Gaizauskas, G. Demetriou, P. J. A., and Willett, P. Protien structures and information extraction from biological texts: The pasta system. *Bioinformatics 19*, 1 (2003), 135143.

21. Sarawagi, S. *Information Extraction*. now Publishers Inc., 2007.

22. Tanabe, L., and Wilbur, W. Tagging gene and protein names in biomedical text. *Bioinformatics 18* (2002), 1124–1132.

23. Tilman, D. Causes, consequences and ethics of biodiversity. *Nature 405* (2000), 208–211.

24. W. W. Chapman, W. Bridewell, P. H. G. F. C., and Buchanan, B. G. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics 34*, 5 (2001), 301310.