# Author name extraction in blog web pages: a machine learning approach

Lucie Dupin[1,2,3], Nicolas Labroche[1], Jean-Yves Antoine[1], Jean-Christophe Lavocat[2], Agata Savary[1]

[1] Université François Rabelais Tours, laboratoire LI, Blois, France – [2] Cicero Labs, Toulon, France – [3] Université Montaigne, Département Sciences du Langage, Bordeaux – France

## Abstract

**Abstract** – This paper presents research results concerning the automatic extraction of author names that are explicitly mentioned in blog web pages. It shows that some NLP pre-preprocessing stages (NE recognition, coreference resolution) prior to a SVM classification have a positive impact on accuracy.

**Résumé** – Cet article présente les résultats de travaux ayant pour but l'extraction automatique de noms d'auteurs explicites dans des articles de blogs. Il montre que l'ajout de pré-traitements relevant du TAL (détection d'entités nommées, résolution des coréférences) avant une classification de type SVM améliore les performances.

**Key words:** author name extraction; blog web pages; machine learning; SVM classifier; decision tree.

## 1. Introduction

Information retrieval from texts receives an increasing attention since Big Data started to be integrated into Web-oriented text mining. The solutions proposed by Natural Language Processing (NLP) aim at retrieving relevant propositional content from electronic documents, but also at conducting understanding-oriented processing of such data (opinion mining for instance). The work presented here focuses on the extraction of the author name of a blog web page, provided that his/her identity is explicitly mentioned. This task significantly differs from the more controversial issue of authorship attribution. We report on experiments which suggest that a combination of an accurate NLP pre-processing and of a supervised classifier lead to a satisfactory performance. We also emphasize the benefits of accurate linguistic features included in the classification process rather than a brute force approach combined with a ranking process.

## 2. Author name extraction

Author name extraction is close to authorship attribution (AA), whose aim is to determine if a document was written by a candidate author whose identity is not revealed in the text. AA applies to plagiarism detection and legal issues. The huge amount of electronic textual data available in the Internet triggered recently a significant change in the paradigm of AA studies: AA systems now massively use machine learning (ML) techniques to identify hidden authors (Statamatos 2009). Standard probabilistic and classification methods are used with a large variety of statistical stylometric features: lexical, character-based, syntactic or even semantic features. Unlike AA, the task of author name extraction (ANE) aims at the identification of a proper name that *explicitly* designates the author in a document. It is therefore not concerned by the sensitive ethical questions that AA raises (Lefeuvre and al. 2015). The existing commercial systems dedicated to ANE are limited to some "harvesting" heuristics on HTML

tags (“author” for instance) while few works have been dedicated ML and NLP-driven approaches. In this paper, we adapt the seminal work of (Changuel et al. 2009) conducted on web pages to a new kind of documents: blog pages.

## 3. Approach

Our approach consists in performing a binary classification (author vs. non-author) on previously identified person named entities (NEs). More precisely, we constructed a processing pipeline shown in figure 1 and described in the next sub-sections.

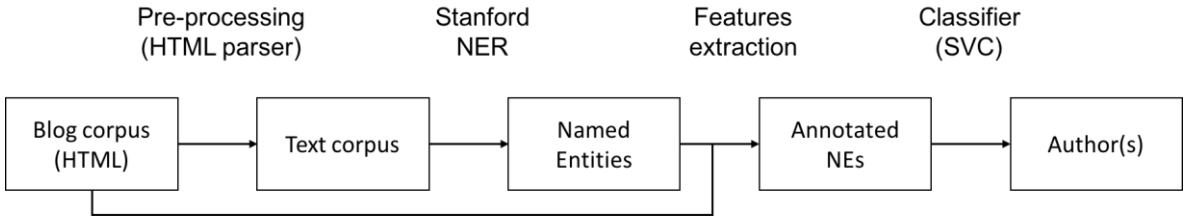


*Figure 1 – Processing pipeline of our author name extraction system.*

### *3.1. Named Entities Recognition*

At first, the HTML files extracted from blog pages are converted into pure textual documents to be processed as raw text by the Stanford Named Entity Recognizer (Finkel and al. 2005). The standard models for English provided with the Stanford NER were directly applied on our data. The best results were obtained with the 4-class (persons, localizations, organizations, others) model trained on CoNLL’2003 data set : `english.conll.4class.distsim.crf.ser`

### *3.2. Machine learning techniques for author detection*

#### *3.2.1. Classification*

There are two main approaches to information extraction (IE) from unstructured data: rule-based systems and ML (Chiticariu, 2014). Rule-based systems are mostly used in industry since they enable a better understanding of the extraction steps. ML approaches are wide spread in research since they enable a more elaborate processing of large datasets without the need for inferring explicit constraints on rules. However, ML still requires the encoding of explicit domain-related knowledge, notably for the definition of features and extraction steps (Kluegl, 2009).

In this work, we focus on ML techniques, which proved efficient in the context of ANE (Changuel et al., 2009). We applied a SVM classifier with a linear SVC kernel, using the *Scikit-learn* platform (http://scikit-learn.org).

ML for IE heavily depends on (i) a choice of features that are discriminative for the desired classification task, and (ii) an algorithm that is suited to the problem and the descriptors. In our formulation of the problem (“*author”* vs. “*non-author”* labelling of pre-identified NEs), there is an additional task of choosing the right author when several person names have been labelled as authors. This problem can occur for two reasons: (i) the document is co-authored by several persons (this case is not covered in our work), (ii) the system has assigned the same likelihood to several name occurrences, corresponding to one or more persons. The latter case can be handled in a post-processing step which consists in re-ranking the candidates (Tako, 2008).

### *3.2.2. Features (descriptors)*

In the literature, ML approaches for entity extraction rely on several types of features. (Freitag, 2000) uses basic features (capitalization, token string, etc.), while others use linguistic ones like part-of-speech, semantic information from gazetteer lists, or NE types (Nadeau, 2006; Amitay, 2004). We have considered 11 binary features (10 descriptive ones + ground truth: feature G). Most of them are related to the document's structure (HTML tags), but we also consider some linguistic clues to depend less heavily on this structure.

Let Doc be the current blog document and E be the current entity under consideration, i.e. the one that is being converted to a set of features. Most features are based on the presence of particular elements in the neighbourhood of E. Two kinds of neighbourhoods are defined. The *textual neighbourhood* (TN) of E is the set of all words located no further than a certain number of words (called the *size* of the TN) from E. This size has been experimentally set to 50. In some cases (see feature V), we use the notion of the *left textual neighbourhood* (LTN), in which only the words to the left of E are included. The size of LTN has been set to 25. The *structural neighbourhood* (SN) relates to the HTML document seen as a tree of HTML elements. The element in which E is most directly embedded is called E's *encapsulating element* (EE). Starting from EE we can follow the branches of the document tree and thus visit EE's descendants, ancestors, siblings, etc. The distance of two elements is understood as the minimum number of branches to be followed in order to get from one of the elements to the other. The structural neighbourhood of E is then the set of all elements whose distance from EE is no higher than a given threshold (called, again, the *size* of the neighbourhood). We empirically set the size of the structural neighbourhood to 7 or 3 (depending on the feature). E's neighbourhoods of size s are denoted TN(E,s), LTN(E,s) and SN(E,s), respectively.

Most opening tags for HTML elements can contain a certain number of attributes. For instance, an element of type <a> can have a @href attribute. We will say that a certain word w *appears in an HTML element* H if w occurs in the value of any attribute of H's opening tag.

*Vocabulary* (Voc) is a set of words (*by*, *about*, *written*, *created*, *vcard*, *updated*, etc.), discovered by a manuel corpus study, which frequently occur in the vicinity of the author's name. We then define a set of binary features each of which is set to 1 if and only if:

- [N1] a date (a string matched by an appropriate regular expression) occurs in TN(E,50).
- [N2] an element of type <img> (*image*) occurs in SN(E,7), *or* the considered NE appears in an element of type <img> anywhere in Doc.
- [V] *by* or *about* occurs in LTN(E,25) *or* any other word from *Voc* occurs in TN(E,50)
- [H1] E's encapsulating element is of type <a>
- [H2] E's encapsulating element EE is of type <a> *and* the word "author" appears in the value of EE's @href attribute
- [H3] E's encapsulating element EE is of type <a> *and* the word "author" appears in the value of any attribute of EE other than @href
- [H4] E's encapsulating element is of type <a> *and* any word from Voc appears in it
- [H5] the word "author" appears in any element from SN(E,3), except EE
- [H6] any word from Voc appears in any element from SN(E,3), except EE

Examples of the trigger elements for features H1 to H6 can be seen in Figure 2 below.

Finally, feature A refers to the merging of multiple name occurrences of the same author in the document. Contrary to (Kato, 2008) that ranks entities to find the most likely author, we

solve the ambiguity by a simple coreference resolution method. We defined several scenarios. In scenario 1, a name re-occurs always under the same form (e.g. *Theodore Roosevelt*): all occurrences are considered to refer to a unique referent. In scenario 2, a name co-occurs with a syntactic variant (*Roosevelt*) for which no competing canonical candidate occurs. Here, we can either consider both occurrences as referring to distinct referents (scenario 2.1) or not (scenario 2.2). In the latter case the longer (*Theodore Roosevelt*) form becomes canonical. In scenario 3, one syntactic variant (*Roosevelt*) has multiple canonical candidates (*Theodore Roosevelt* vs. *Franklin Roosevelt*). Then, we can keep the ambiguity unresolved (scenario 3.1) or apply a brute force approach (scenario 3.2) considering the ambiguous form *Roosevelt* as coreferent with both canonical candidates. Feature A is set to 1 (merging) in the scenarios 1, 2.2 and 3.2. In such merging scenario, only the canonical form is retained: its features result from the disjunction of the features of both entities to be merged as in (Changuel, 2009).

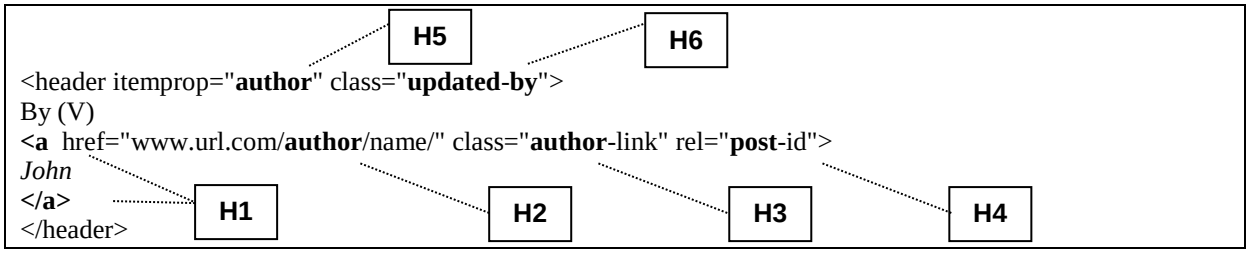


*Figure 2 – Feature assignment for the named entity "John" in a fake HTML document*

## 4. Results

The experiments were conducted on two corpora from two different English blog domains. The first corpus (*base*), divided into two parts – the training part (*base-train:* 600 English blog pages) and the testing part (*base-test*: 100 additional pages) – concerns a unique blog domain. The second one (*inc*) was created to assess the systems on a different domain. All performances were evaluated in terms of accuracy (% of discovered authors). The experiments reported here were notably meant to investigate the benefits of the addition of NLP-based considerations in a standard classification process. Namely, we assessed several configurations of the system representing combinations of the following options:

- All the extracted NEs regardless of their type (-PERS), or only those of type "person" (+PERS) were retained.
- Lexical feature V (*vocabulary*) was (+V) or was not (-V) taken into account.
- Concerning feature A, name variants were never merged (-M), they were merged (+M+A) only in case of no ambiguity (scenarios 2.2 and 3.1) or they were always merged.

**Influence of named entities categorization** – The first experiment investigates the benefits of introducing the categorization of NEs in the preprocessing stage prior to the classification. We selected the configurations +V and +M+A and combined them with -PERS on the one hand and with +PERS on the other hand. Retaining only the extracted person names leads to a significant increase in the performances of the system (Table 1): the `+PERS+V+M+A` configuration succeeds in identifying 91% of the author names, while the system without this NLP preprocessing obtains an accuracy of 78%.

| System | **-PERS+V+M+A** | **+PERS+V+M+A** |
|---|---|---|
| Accuracy (% of author names correctly identified) | 0.78 | 0.91 |

*Table 1. Influence of retaining personal named entities only on accuracy*

**Influence of merging coreferent person names** – The second stage of NLP preprocessing consists in merging explicitly coreferent NEs. Three systems are compared: merging of unambiguous names (`+PERS+V+M+A`), ambiguous merging (`+PERS+V+M-A`) and no merging (`+PERS+V-M`). Table 2 shows that the merging of the explicitly coreferent entities (`+M-A` or `+M+A`) is essential to obtain satisfactory performances. It demonstrates that a correct identification of the author should not be based on local decisions of the classifier, but rather on a NLP determination of sets of coreferent entities. Unsurprisingly, avoiding ambiguous merges (`PERS+V+M+A` system) achieves the best accuracy measure.

| System | **+PERS+V-M** | **+PERS+V+M-A** | **+PERS+V+M+A** |
|---|---|---|---|
| Accuracy | 0.38 | 0.36 | 0.91 |

*Table 2. Influence of merging coreferent person names prior to the classification*

**Generalization on any blog domain: influence of a task-specific vocabulary** – The last experiment compares two systems in which the vocabulary is used (`+PERS+V+M+A`) or not (`+PERS-V+M+A`). Our hypothesis is that this lexical feature should lead to a better cross-domain scalability of the system. For that purpose, the systems were trained on the `base-train` corpus, and tested on the `base-test` and `inc` corpora. The results presented in Table 3 are quite disappointing: the influence of the V feature is restricted and we observe a significant decrease of the accuracy on the out-of-domain evaluation corpus (*inc*). This lack of generalization power must be investigated in the future

| System | **+PERS-V+M+A** | **+PERS+V+M+A** |
|---|---|---|
| Accuracy: *Base-test* corpus | 0.91 | 0.91 |
| Accuracy: *Inc* corpus | 0.66 | 0.65 |

*Table 3. Influence of the vocabulary-based feature on the classification*

## References

Amitay, E. and al. (2004) Web-a-where: geotagging web content. *Proc. of SIGIR 04*. Pages 273-280.

Breiman L., Friedman J., Olshen R., Stone C. (1984) *Classification and Regression Trees*.

Changuel S., Labroche N., Bouchon-Meunier B. (2009) Automatic Web Pages Author Extraction. *Proc. Of the FQAS 2009 conference*. Pages 300-311.

Chiticariu L., Li Y., Reiss F.R. (2013) Rule-based Information Extraction is Dead ! Long Live Rule-based Information Extraction Systems. *Proc. EMLP'2013*. Pages 827–832.

Finkel J.R., Grenager T., Manning C. (2005) Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proc. ACL 2005*, pages 363-370

Freitag, D., Kushmerick, N. (2000) Boosted wrapper induction. *Proc. CIAA'2000*. Pages 577-583.

Kato, Y., Kawahara, D., Inui, K., Kurohashi, S., Shibata, T. (2008) Extracting the author of web pages. *Proc. of the 2nd ACM workshop on Inf. credibility on the web, WICOW 08*. Pages 35-42.

Kluegl P., Atzmueller M., Puppe F. (2009) TextMarker: A Tool for Rule-Based Information Extraction. *Proc. of the GSCL Conference 2009, 2nd UIMA@GSCL Workshop*. Pages 233-240.

Lefeuvre A., Antoine J.-Y., Allegre W. (2015) Ethique conséquentialiste et traitement automatique des langues, *Atelier ETeRNAL'2015, TALN'2015*. Pages 53-66.

Nadeau, D., Turney, P., Matwin, S. (2006) Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. *Canadian AI 2006*. Pages 266-277.

Statamatos E. (2009) A Survey of Modern Authorship Attribution Methods. *Journal of the American Society for Information Science and Technology*, 60(3), pages 538-556.