-

Mining the Web for lists of Named Entities

Arlind Kopliku — Mohand Boughanem — Karen Pinel-Sauvagnat

Institut de Recherche en Informatique de Toulouse, UMR 5505 CNRS, SIG-RFI, 118 route de Narbonne F-31062 Toulouse Cedex 9, France
Arlind.Kopliku@irit.fr, Mohand.Boughanem@irit.fr, Karen.Sauvagnat@irit.fr

ABSTRACT. *Named entities play an important role in Information Extraction. They represent unitary namable information within text. In this work, we focus on groups of named entities of the same type which we try to extract from HTML lists. Instead of starting from a class and identifying the corresponding named entities, we want to explore a new paradigm which consists in identifying sets of named entities without any knowledge on the class. A clear advantage of the approach is that it is applicable to all named entities (no matter what class), which makes it domain independent. We use HTML lists to collect candidate sets of named entities. Human assessors assessed a randomly selected sample of HTML lists. 8,25% of these HTML lists are lists of named entities of the same class. If our estimation is validated at large scale, it is possible to expect at least 890 million of such lists of named entities only in the indexed Web. Moreover, we propose an appropriate classifier which shows promising results.*

RÉSUMÉ. *Les entités nommées jouent un rôle important en extraction d'information. Dans cet article, nous proposons une méthode pour extraire des entités nommées de la même classe au sein de listes HTML. Au lieu de partir d'une classe donnée et d'extraire les entités correspondantes, nous proposons une nouvelle approche qui consiste à identifier des ensembles d'entités nommées sans connaître leur classe d'appartenance. Un avantage évident de cette approche est qu'elle peut s'appliquer à tout type d'entité nommée (c'est à dire à des entités nommées de n'importe quelle classe). Nous utilisons des listes HTML pour identifier des ensembles candidats d'entités. Afin d'évaluer notre approche, des juges ont évalué un échantillon de listes HTML issues du Web. 8.25% de ces listes sont des listes d'entités nommées de la même classe. On peut ainsi s'attendre à trouver plus de 890 millions de listes d'entités nommées appartenant à la même classe sur tout le Web indexé. Le classifieur que nous proposons dans cet article et permettant d'identifier ces listes d'entités nommées pertinentes nous permet d'obtenir de premiers résultats prometteurs.*

KEYWORDS: *Information Extraction, named entities, HTML list, Information Retrieval*

1. Introduction

Named entities are frequent in texts. Each named entity has at least one class and different named entities can relate with each other. Their identification allows to add some semantics to the content. Typical classes of named entities are persons, organization, locations such as John Travolta, Sun Microsystems, France. They can relate to each other such as in the phrase *"John Travolta"* played a role in *"Pulp Fiction"*.
Identifying the correct class is not an easy task and for some classes it is easier than for others. There are some important issues worth mentioning. The same name can refer to different entities (Bunescu *et al.*, 2006) (e.g Mr.Jones). The same named entity can belong to several classes (Hearst, 1992; Snow *et al.*, 2006) (e.g Hotel California can match a song and a hotel). Moreover, it is not easy to agree on a common taxonomy even for a single language, as well as it is impossible to list all possible classes.
In this work, we separate the instances from their class to see if it is possible to extract some other semantics and utility from named entities. We interest in sets of named entities of common but anonymous class, which we call *siblings' set* or *siblings' lists*. The goal of this work is an automatic extraction of siblings' sets, which is domain independent and applicable at large scale. Siblings sets can be an alternative representation of extraed information. They can be used for set expansion (Wang *et al.*, 2008). They can be later used to answer qualified list queries (Cafarella *et al.*, 2006) such as "US presidents", "Milan FC players who are not Italian", "UE members", etc. Having a siblings' set helps disambiguate the class of the named entities in it. For instance, the set "Hotel California, Desperado, Lying eyes" is more likely to be songs than hotels.
We choose HTML lists as candidates for extracting siblings' lists for some reasons. First, HTML lists can be used for qualified lists. Second, HTML lists are easy to parse. Third, there are many HTML lists in the Web. Nevertheless, we take into account that lists are often used for navigation menus, layout design, etc.
Our contribution can be summarized with these points. First, we introduce the notion of siblings' set. Second, we analyze HTML lists for a massive and domain-independent extraction of siblings' sets. Third, we implement an ad-hoc classifier.
This paper is structured as follows. Next section is about related work. Then, we describe the experimental setup and the results. The last section is about conclusions and future work.

2. Related work

Named entity Recognition can identify class instances only for a subset of classes such as person names, locations, organizations (Grishman *et al.*, 1996; Sekine *et al.*, 2002; Etzioni *et al.*, 2005). Many approaches are class-specific and they cannot be extended to all classes.
A class independent technique to extract instances was identified by Hearst in (Hearst, 1992). He uses a simple lexico-syntactic rule concretely "C such as LI". This rule matches with "countries such as Spain, Italy and France", "cities such as London and Tokyo". This technique as many others cannot identify all possible denominations

of a class. The set of instances "Spain, Italy and France" can be associated to "Latin language speaking countries", but also "FIFA 2010 qualified teams", "European countries", "EU members". The identification of the class might not be easy. Removing this constraint can increase the number of extractable named entities and enable new uses.
There is other work considering semi-structured content within HTML pages. Cohen, Hurst, Jensen (Cohen *et al.*, 2002) learn wrappers for lists and tables, while Agichtein and Gravano (Agichtein *et al.*, 2000) focus on wrappers for HTML tables. Our work differs from theirs as we use a simple wrapper for lists, which works well without aiming the best wrapper. Our focus is on the quality of the extracted lists.

3. The experimental setup

3.1. *Extraction of lists*

We consider only HTML lists i.e. lists within the respective HTML tags <OL>, <UL> and <DL>. Within these lists, we filter out the ones which were probably useless. For some cases, it is evident why the list cannot be a siblings' set, while for some others it was necessary to analyze data in advance. Concretely, the following lists are filtered out: lists without end tags, lists with empty items, lists with less than 3 items, lists with items of improbable length in characters [1], lists with an imprabable average length in characters [2].
Analyzing the parsed documents, we find about 3.4 lists per page and about 5.7 items per list. We also notice that the length of the lists varies a lot, where the longest list has 466 items. Filtering removes 6 lists out of 7. However, the remaining dataset is still huge enough to extract meaningful results.

3.2. *Evaluation*

Lists are extracted from the QUAERO [3] document corpus which contains French language pages crawled from the Web. Evaluation is done by 5 PhD students over a subset of 2000 lists chosen randomly. He had to tell, if a given list is a siblings' set of named entities of the same class or not. The assessor could access the source document of the list and a search engine when needed.
The assessor was also told that the named entity should be the entire list item and not simply present in the item. For lists longer than 5, one wrong element was marked as acceptable (e.g. France, Germany, Italy, Spain, Albania, Others). Assessments were also used to analyze the distribution of siblings' lists, named entities and to detect potentially discriminating features.

[1] less than 3 characters, longer than 40 characters

[2] less than 4 characters, longer than 32 characters

[3] QUAERO is a project among French and German public and private research organizations (http://www.quaero.org)

4. Results and case studies

From 2000 assessed lists only 165 of them were judged as lists of named entities of the same type. This is only 8.25% of them. If we consider that we filtered 6 out of 7 lists and that there are 3.4 lists per page, it is possible to estimate for our dataset that there are at least 3.99 siblings' lists each hundred documents. This is a small proportion, but it is a strong indicator that the Web with its size is a rich mine for extracting siblings' lists. More concretely, if we consider an estimation of the number of indexed pages in 2009 which is 22.3 billions[4], we can expect to extract about 892,000,000 siblings' lists which is huge and would be probably the largest collection of useful lists ever collected.

To validate this estimation, it might be necessary to repeat the same experiment with a larger dataset, at least one order of magnitude. As well, it will be interesting to check whether language and cultural differences affect the use of lists and their quality.

Below we consider some important features which can help to detect subsets with larger percentage of siblings' lists. This is useful to understand which features are discriminative and for future work to design appropriate good classifiers.

4.1. *Links (anchors)*

Many Web Designers often use lists to show navigation options (menus), to propose related links or simply for advertisement links. This is very frequent and one can expect to find fewer lists of named entities within these cases. It is common to expect links (anchors) within the item of such lists. This is why we studied the influence of links in the quality of lists.

We consider 3 cases. The first one is lists where all items have links ("all links" case). The second case is lists where at least one item has no links ("some links" case). In the third case, we consider only lists which do not have any link at all ("no links" case).

The result are shown in table 1. The first column shows the case. The second column shows how many lists from our filtered sample share fall in each class, followed by a percentage relative to the size of the sample. The percentages are useful to understand how frequent lists of each type are. The results show that the lists where all items have links are less probable to be siblings' sets. Among lists with at least one item which

Table 1. *Quality and number of lists for link related features*

Set	Siblings' lists %	Number of lists
All lists	8.2%	59756
All links	7.5%	53441 (85.6%)
Some links	17.5%	6345 (10.6%)
No links	22.5%	1750 (2.9%)

[4] Taken from http://www.worldwidewebsize.com/ on October 2009

is not linked there 17.5% which are siblings' lists, while if there are no links at all the results are even better (22.5%).
Although, among lists with all items with links only 7.5% are siblings' lists, we cannot neglect this subset. They represent about 85.6% of the lists of our list dataset, which makes this set a very important seed to find siblings' lists.

4.2. *The item frequencies*

In analogy with term frequency in a document (*tf*) or the document frequency (*df*), we define the item frequency (*if*) for lists as in how many lists an item is found in the lists of the collection. We study if such a frequency has a role in the quality of lists. For instance menu items can be very frequent such as *"Home"*.
Table 2 shows the items which are the most frequent in lists. As one can see, these are mostly items that help navigation, but not named entities. In fact, most of them appear mostly with links.
We observed that if the list has an item which repeats very frequently with links, the list is less probable to be qualitative. For instance, the percentage of good lists is only 3.2% for lists where there is an item with a frequency (*if*) above 160 times[5]. Furthermore, we observe that among lists with an average item frequency above 40 times[6], only 0.7% are qualitative.
There are surprisingly a lot of lists with a maximal item frequency above 160 (24273 lists, about 40.6%) and average item frequency above 40 (27131 lists, about 45.4%). The above statistics can be very useful to filter out potentially useless lists leaving a subcollection with almost two times better quality.
On the other hand, focusing on lists for which all items appear only once, we found 40% of siblings' lists, which is much above the dataset's average.

Table 2. *The most frequent items in the lists of our dataset*

Name	English translation	Frequency
Accueil	Home	3887
Forum	Forum	1819
Contact	Contact	1506
Recherche	Search	1448

4.3. *Ordered versus unordered lists*

We could observe that there are much more unordered lists in the web than ordered ones. Only 0.4% of the entire set are ordered. However, about 60% of the latter are

[5] Chosen by observations
[6] Chosen by observations

siblings' lists. We can state that the quality of ordered lists is much higher than for unordered lists. This can be explained by the fact that ordered lists are closer to the traditional meaning of list.

4.4. *The number of items per list*

Another feature we studied is the influence of the number of items. Are short lists more probable to contain named entities of the same type or longer ones behave better? Analyzing results, we could distinguish 4 ranges that behave differently. The range 3-6 has an average quality of 6.8%. The range 7-9 has an average quality of 12.5%. The range 10-15 has an average quality of 4%, while if there are more than 15 items the average quality is 15%. The results tell that there are twice as many quality lists in the range 7-9 with respect to the range 3-6. It also tells that there are more quality lists for long lists (more than 15 items).

5. Classification

In this section we propose a binary classification for HTML lists, which has to detect if a list is a siblings' set or not. The feature we use are listed in table 3. The values are transformed into real numbers and normalized when necessary.

Table 3. *The features used for classification*

Feature	Description	Feature	Description
Type	Is the list ordered?	AvgLgth	Avg. item length in characters
Length	No. of items of the list	Longest	Max. item length in characters
MinIF	Min. item frequency	NoLinks	No list items with anchors
MaxIF	Max. item frequency	AllLinks	All list items have anchors
AvgIF	Avg. item frequency		

We use the open source LIBSVM library (Chang *et al.*, 2001), which provides support vector classification, (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM). These three types of classification can be used with 4 types of kernels namely the radial based function, the sigmoid function, linear combination and polynomial combination.

We tested all three types of SVM classifiers with the 4 kernels. In each case, different combinations of features were also input to see if there are feature combinations which are more discriminative than others.

To validate results a 3 folded cross-validation is used. This consists in dividing the dataset in 3 subsets, where 2 are used for training and 1 for testing. The choice of the test set is then altered three times and the classifiers performance is better estimated by an average of the three turns. As measures, we use precision and recall for both the positive class (siblings' set) and negative set (other lists).

It is impossible to show all the obtained results for all combinations of features and configurations. In table 4, we show 4 interesting cases to compare. We take a *always-yes* answering classifier and an *always-no* answering classifier. We put them aside with two well performing classifiers, one that performs particularly well in precision (C-SVC) and another that obtains good recall at the cost of precision loss (ONE-CLASS).

Table 4. *Precision and recall of the classification*

	Positive		Negative	
Classifier	**Prec.**	**Recall**	**Prec.**	**Recall**
Always Yes	8.25%	100%	-	0%
Always No	-	0%	91.75%	100%
C-SVC [15]	57.1%	18.1%	91.5%	98.4%
ONE-CLASS [16]	14.0%	44.7%	92.2%	71.0%

The best results were obtained with the radial based function and the most discriminative features are found to be the presence of links, the type of list (ordered or not) and the item frequency. Our results vary in terms of precision and recall. With a 100% of recall precision is low. Reasonable results for precision and recall are obtained respectively with the third (C-SVC) and the forth classifier (ONE-CLASS).
The results remain encouraging as we address a domain independent massive extraction of lists of named entities of the same type. We also believe that the introduction of new features and especially the PMI measure will significantly improve results. These results show the benefits and limits of HTML list specific features.

6. Conclusions and future work

Named entities, their classes and relations have long been studied. In this paper we propose an alternative approach for named entity extraction where we leave the class identification as a separate step. We argue that groups of named entities of the similar type have multiple uses and they can be an alternative organization for named entities which can be applied to all named entities.
We use HTML lists as candidates to identify sets of named entities of the same class, which we call siblings' sets. We show that about 8.25% of the HTML lists in our dataset are siblings' sets. If our estimation is validated and can be generalized to the Web, using an estimation of the size of the Web, we can estimate to have more 892 million lists of named entities of the same type, which corresponds to billions of named entities.
The identification of such a huge collection of named entities of the same type is

[15] Features considered: all

[16] Features considered: presence of links, the type of list (ordered or not) and the item frequency

only one part of the contribution. Further analysis on our assessed lists is used to understand which features are more discriminative for siblings' sets.
The observations are used to build automatic binary classification to detect lists of named entities. The performance of the classifier varies with respect to the features used and the chosen configuration. We obtain 57.1% of precision and 18.1% of recall as well as 14% of precision and 44.7% recall. Depending on the uses the classifier can be tuned to bias recall or precision. HTML list features are shown to be discriminative, but they have their limits. We believe that the integration of new features will further improve results. Still, these are promising results for a large scale extraction of siblings' lists.

7. References

Agichtein E., Gravano L., "Snowball: extracting relations from large plain-text collections", *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, ACM, New York, NY, USA, pp. 85–94, 2000.

Bunescu R. C., Pasca M., "Using Encyclopedic Knowledge for Named entity Disambiguation", *EACL*, The Association for Computer Linguistics, 2006.

Cafarella M. J., Banko M., Etzioni O., Relational Web Search, Technical report, University of Washington, 2006.

Chang C.-C., Lin C.-J., *LIBSVM: a library for support vector machines*. 2001.

Cohen W. W., Hurst M., Jensen L. S., "A flexible learning system for wrapping tables and lists in HTML documents", *WWW '02: Proceedings of the 11th international conference on World Wide Web*, ACM, New York, NY, USA, pp. 232–241, 2002.

Etzioni O., Cafarella M., Downey D., Popescu A.-M., Shaked T., Soderland S., Weld D. S., Yates A., "Unsupervised named-entity extraction from the Web: an experimental study", *Artif. Intell.*, vol. 165, num. 1, pp. 91–134, 2005.

Grishman R., Sundheim B., "Message Understanding Conference-6: a brief history", *Proceedings of the 16th conference on Computational linguistics*, Association for Computational Linguistics, Morristown, NJ, USA, pp. 466–471, 1996.

Hearst M. A., "Automatic acquisition of hyponyms from large text corpora", *Proceedings of the 14th conference on Computational linguistics*, Association for Computational Linguistics, Morristown, NJ, USA, pp. 539–545, 1992.

Sekine S., Sudo K., Nobata C., "Extended Named Entity Hierarchy", *Proceedings of LREC 2002*, 2002.

Snow R., Jurafsky D., Ng A. Y., "Semantic taxonomy induction from heterogenous evidence", *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Morristown, NJ, USA, pp. 801–808, 2006.

Wang R. C., Cohen W. W., "Iterative Set Expansion of Named Entities Using the Web", *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, IEEE Computer Society, Washington, DC, USA, pp. 1091–1096, 2008.