Combining Lexical and Formatting Cues for Named Entity Acquisition from the Web

Christian Jacquemin[1] **and Caroline Bush**[1,2]
[1]CNRS-LIMSI, BP 133, F-91403 ORSAY Cedex, FRANCE
[2]UMIST, Dept of Language Engineering, PO Box 88, Manchester M60 1QD, UK
{jacquemin,caroline}@limsi.fr

Abstract

Because of their constant renewal, it is necessary to acquire fresh named entities (NEs) from recent text sources. We present a tool for the acquisition and the typing of NEs from the Web that associates a harvester and three parallel shallow parsers dedicated to specific structures (lists, enumerations, and anchors). The parsers combine lexical indices such as discourse markers with formatting instructions (HTML tags) for analyzing enumerations and associated initializers.

1 Overview

Lexical acquisition from large corpora has long been considered as a means for enriching vocabularies (Boguraev and Pustejovsky, 1996). Depending on the studies, different issues are considered: the acquisition of terms (Daille, 1996), the acquisition of subcategorization frames (Basili et al., 1993), the acquisition of semantic links (Grefenstette, 1994), etc. While traditional electronic corpora such as journal articles or corpus resources (BNC, SUSANNE, Brown corpus) are satisfactory for classical lexical acquisition, Web corpora are another source of knowledge (Crimmins et al., 1999) that can be used to acquire NEs because of the constant updating of online data.

The purpose of our work is to propose a technique for the extraction of NEs from the Web through the combination of a harvester and shallow parsers. Our study also belongs to corpus-based acquisition of semantic relationships through the analysis of specific lexico-syntactic contexts (Hearst, 1998) because hypernym relationships are acquired together with NEs. The unique contribution of our technique is to offer an integrated approach to the analysis of HTML documents that associates lexical cues with formatting instructions in a single and cohesive framework. The combination of structural information and linguistic patterns is also found in *wrapper induction*, an emerging topic of research in artificial intelligence and machine learning (Kushmerick et al., 1997).

Our work differs from the MUC-related NE tagging task and its possible extension to name indexing of web pages (Aone et al., 1997) for the following reasons:

- The purpose of our task is to build lists of NEs, not to tag corpora. For this reason, we only collect non-ambiguous context-independent NEs; partial or incomplete occurrences such as anaphora are considered as incorrect.
- The types of NEs collected here are much more accurate than the four basic types defined in MUC. The proposed technique could be extended to the collection of any non-MUC names which can be grouped under a common hypernym: botanic names, mechanical parts, book titles, events...
- We emphasize the role of document structure in web-based collection.

2 Focusing on Definitory Contexts

Two issues are addressed in this paper:

1. While traditional electronic corpora can be accessed directly and entirely through large-scale filters such as shallow parsers, access to Web pages is restricted to the narrow and specialized medium of a search engine. In order to spot and retrieve relevant text chunks, we must focus on linguistic cues that can be used to access pages containing typed NEs with high precision.
2. While Web pages are full of NEs, only a small proportion of them are relevant for the acquisition of public, fresh and well-known NEs (the name of someone's cat

is not relevant to our purpose). So that automatically acquired NEs can be used in a NE recognition task, they are associated with types such as *actor* (PERSON), *lake* (LOCATION), or *university* (ORGANIZATION).

The need for selective linguistic cues (wrt to the current facilities offered by search engines) and for informative and typifying contexts has led us to focus on collections, a specific type of definitory contexts (Péry-Woodley, 1998). Because they contain specific linguistic triggers such as *following* or *such as*, definitory contexts can be accessed through phrase queries to a search engine. In addition, these contexts use the classical scheme *genus/differentia* to define NEs, and thus provide, through the *genus*, a hypernym of the NEs they define. Our study extends (Hearst, 1998) to Web-based and spatially formatted corpora.

3 Architecture and Principles

To acquire NEs from the Web, we have developed a system that consists of three sequential modules (see Figure 1):

1. A harvester that downloads the pages retrieved by a search engine from the four following query strings

(1.a) *following* ⟨NE⟩ (1.c) ⟨NE⟩ *such as*
(1.b) *list of* ⟨NE⟩ (1.d) *such* ⟨NE⟩ *as*

   in which ⟨NE⟩ stands for a typifying hypernym of NEs such as *Universities*, *politicians*, or *car makers* (see list in 4).

2. Three parallel shallow parsers `Pe`, `Pl` and `Pa` which extract candidate NEs respectively from enumerations, lists and tables, and anchors.

3. A post-filtering module that cleans up the candidate NEs from leading determiners or trailing unrelated words and splits coordinated NEs into unitary items.

Corpus Harvesting

The four strings (1.a-d) given above are used to query a search engine. They consist of an hypernym and a discourse marker. They are expected to be followed by a collection of NEs.

Figure 2 shows five prototypical examples of collections encountered in HTML pages retrieved through one of the strings (1.a-d).[1] The first collection is an **enumeration** and consists of a coordination of three NEs. The second collection is a **list** organized into two sublists. Each sublist is introduced by a hypernym. The third structure is a **list** marked by bullets. Such lists can be constructed through an HTML table (this example), or by using enumeration marks (<*ul*> or <*ol*>). The fourth example is also a **list** built by using a table structure but displays a more complex spatial organization and does not employ graphical bullets. The fifth example is an **anchor** to a collection not provided to the reader within the document, but which can be reached by following an hyperlink instead.

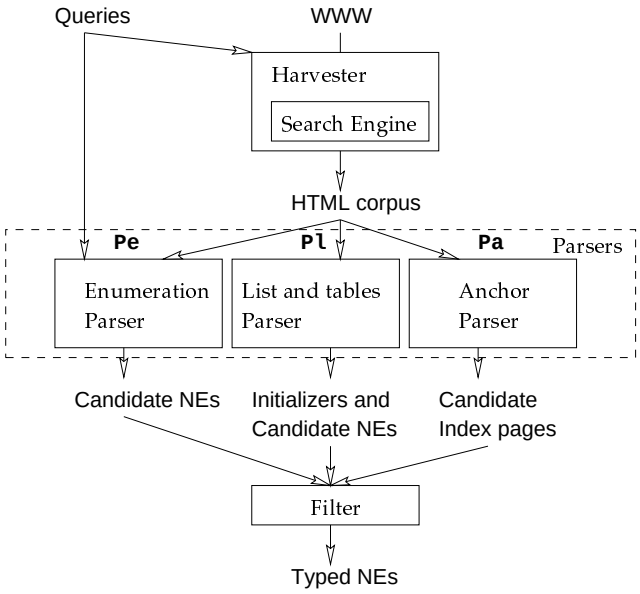

Figure 1: Architecture

The corpus of HTML pages is collected through two search engines with different capabilities: AltaVista (AV) and Northern Light (NL).[2] AV offers the facility of double-quoting the query strings in order to search for exact strings (also called phrases in IR). NL does not support phrase search. However, in AV, the number of retrievable documents is limited to the 200 highest ranked documents while it is potentially unlimited in NL. For NL, the

[1] ⟨NE⟩ is *international organizations*, here. The typographical mark-up of the query string in the figure is ours. The hypernym is in bold italics and the discourse marker is in bold.

[2] The harvester that retrieves Web pages through a search engine is a combination of `wget` available from `ftp://sunsite.auc.dk/pub/infosystems/wget/` and Perl scripts.

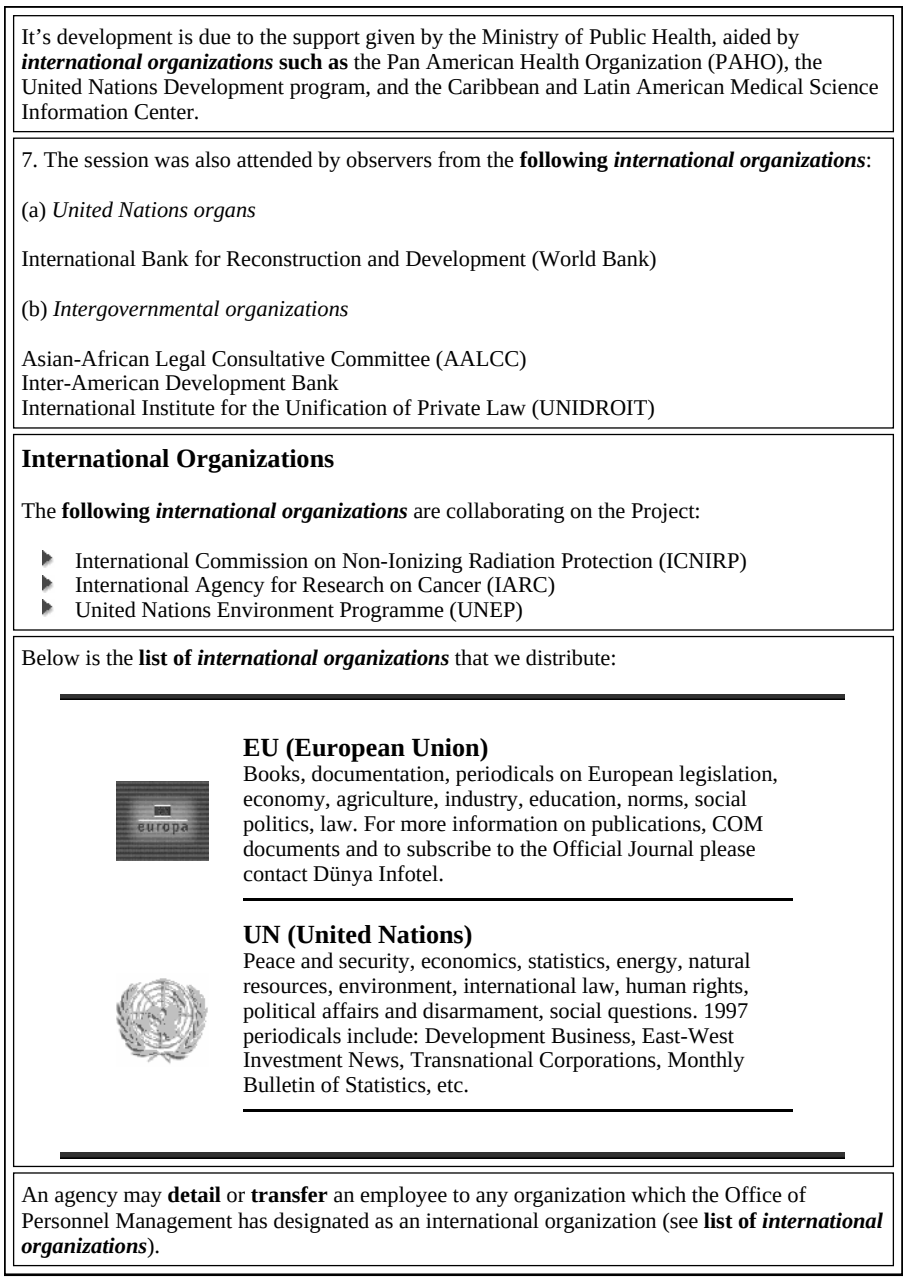

Figure 2: Five different types of formatting used for enumerating NEs.

number of retrieved documents was however restricted to 2000 in order to limit processing times. The choice of these two search engines is intended to evaluate whether a poorer query mode (bags of words in NL instead of strings in AV) can be palliated by accessing more documents (2000 max. for NL instead of 200 max. for AV).

The corpus collected by the two search engines and the four families of queries is 2,958Mb large (details are given in Section 4).

Acquisition of Candidate NEs

Three parallel shallow parsers `Pe`, `Pl` and `Pa` are used to extract NEs from the corpora collected by the harvester. The parsers rely on the query string to detect the sentence introducing the collection of NEs (the *initializer* in (Péry-Woodley, 1998)). The text and HTML marks after the initializer are parsed jointly in order to retrieve one of the following three spatio-syntactic structures:

1. a textual enumeration (parser `Pe`, top-

most example in Figure 2),

2. a list or a table (parser `Pl`, the next three examples in Figure 2),
3. an anchor toward a page containing a list (parser `Pa`, bottom example in Figure 2).

In brief, these parsers combine string matching (the initial lexical cue), syntactic analysis (enumerations in `Pe`), analysis of formatting instructions (lists and tables in `Pl`), and access to linked documents through anchors detected by `Pa`. The results presented in this paper only concern the first two parsers. Since anchors raise specific problems in linguistic analysis (Amitay, 1999), they will be analyzed in another publication. The resulting candidate NEs are cleaned up and filtered by a post-filtering module that splits associations of NEs, suppresses initial determiners or trailing modifiers and punctuations, and rejects incorrect NEs.

The Enumeration Parser `Pe`

The enumerations are expected to occur inside the sentences containing the query string. `Pe` uses a traditional approach to parsing through conjunction splitting in which a NE pattern NE is given by (3) and an enumeration by (4).[3]

$$NE = ([A\text{-}Z \setminus \&][a\text{-}zA\text{-}Z \setminus \text{-}\setminus']^{\star})^{+} \quad (3)$$

$$Enum = (NE,\ )^{\star}\, NE\, (,^{?})\ (and|or)\ NE \quad (4)$$

The List Parser `Pl`

The lists are expected to occur no further than four lines after the sentence containing the query string. The lists are extracted through one of the following three patterns. They correspond to three alternatives commonly used by HTML authors in order to build a spatial construction of aligned items (lists, line breaks, or tables). They are expressed by case-insensitive regular expressions in which the selected string is the **shortest** acceptable underlined pattern:

$$ \ \underline{.^{\star}}\ (| | |) \quad (5)$$

$$
\ \underline{.^{\star}}\ </br> \quad (6)$$

$$(<td>|<th>)\ \underline{.^{\star}}\ (<td>|<th>|</td>|</th>|</table>) \quad (7)$$

[3]The patterns are slightly more complicated in order to accept diacriticized letters, and possible abbreviations composed of a single letter followed by a dot.

In addition, after the removal of the HTML mark-up tags, only the **longest** subpart of the string accepted by (3) is produced as output to the final filter. These patterns do not cover all the situations in which a formatted text denotes a list. Some specific cases of lists such as pre-formatted text in a verbatim environment (<*pre*>), or items marked by a paragraph tag (<*p*>) are not considered here. They would produce too inaccurate results because they are not typical enough of lists.

Postfiltering

The pre-candidate NEs produced by the shallow parsers are processed by filters before being proposed as candidate NEs. The roles of the filters are (in this order):

- removal of trailing lower-case words,
- deletion of the determiner *the* and the coordinating conjunctions *and* and *or* and the words which follow them,
- rejection of pre-candidates that contain the characters `@`, {, # , ~, $, ! or ?.
- suppression of item marks such as 1., —, ∗ or a),
- suppression of HTML markups,
- suppression of leading coordinating conjunctions,
- suppression of appositive sequences after a comma or a hyphen,
- transformation of upper-case words into initial upper-case in non-organization candidate NEs because only organization names are expected to contain acronyms,
- rejection of NEs containing words in a stop list such as *Next*, *Top*, *Web*, or *Click*.

Postfiltering is completed by discarding single-word candidates, that are described as common words in the CELEX[4] database, and multi-word candidates that contain more than 5 words.

4 Experiments and Evaluations

Data Collection

The acquisition of NEs is performed on 34 types of NEs chosen arbitrarily among three subtypes of the MUC typology:

[4]The CELEX database for the English language is available from the Consortium for Lexical Resources at `www.ldc.upenn.edu/readme_files/celex.readme.html`.

ORGANIZATION (*American companies*, *international organizations*, *universities*, *political organizations*, *international agencies*, *car makers*, *terrorist groups*, *financial institutions*, *museums*, *international companies*, *holdings*, *sects*, and *realtors*),

PERSON (*politicians*, *VIPs*, *actors*, *managers*, *celebrities*, *actresses*, *athletes*, *authors*, *film directors*, *top models*, *musicians*, *singers*, and *journalists*), and

LOCATION (*countries*, *regions*, *states*, *lakes*, *cities*, *rivers*, *mountains*, and *islands*).

Each of these 34 types (a ⟨NE⟩ string) is combined with the four discourse markers given in (1.a-d), yielding 136 queries for the two search engines. Each of the 272 corpora collected through the harvester is made of the 200 documents downloadable through AV for the phrase search (or less if less are retrieved) and 2,000 documents though NL. Each of these corpora is parsed by the enumeration and the list parsers.

Two aspects of the data are evaluated. First, the size of the yield is measured in order to compare the productivity of the 272 queries according to the type of query (type of NE and type of discourse marker) and the type of search engine (rich versus plain queries and low versus high number of downloaded documents). Second, the quality of the candidate NEs is measured through human inspection of accessible Web pages containing each NE.

Corpus Size

The 272 corpora are 2,958 Mb large: 368 Mb for the corpora collected through AV and 2,590 Mb for those obtained through NL. Detailed sizes of corpora are shown in Table 1. The corpora collected through NL for the pattern *list of* ⟨NE⟩ represent more than a half of the NL collection (1,307 Mb). The most productive pattern for AV is ⟨NE⟩ *such as* through which 41% of the AV collection is downloaded (150 Mb).

The sizes of the corpora also depends on the type of NEs. For each search engine, the total sizes are reported for each pattern (1.a-d). In addition, the largest corpus for each of the three types of NEs is indicated in the last three lines. The variety of sizes and distribution among the types of NEs shows that using search engines with different capabilities yields different figures for the collections of pages. Therefore, the subsequent process of NE acquisition heavily depends on the means used to collect the basic textual data from which knowledge is acquired.

Quantitative Evaluation of Acquisition

Table 2 presents, for each pattern and each search engine, the number of candidates, the *productivity*, the *ratios of the number of enumerations to lists*, and the rate of *redundancy*.

In all, 17,176 candidates are produced through AV and 34,978 through NL. The lowest accuracy of the NL query mode is well palliated by a larger collection of pages.

Productivity. The productivity is the ratio of the number of candidates to the size of the collection. Using a unit of number of candidates per Mb, the productivity of AV is 46.7 while it is 3.5 times lower for NL (13.5). Thus, collecting NEs from a coarser search engine, such as NL, requires downloading 3.5 times larger corpora for the same yield. A finer search engine with phrase query facilities, such as AV, is more economical with respect to knowledge acquisition based on discourse markers.

As was the case for the size of the collection, the productivity of the corpora also depends on the types of NEs. *Universities* (28.1), *celebrities* (53.0) and *countries* (36.5) are the most productive NEs in their categories while *international agencies* (4.0), *film directors* (4.4) and *states* (8.7) are the less productive ones. These discrepancies certainly depend on the number of existing names in these categories. For instance, there are many more names of *celebrities* than *film directors*. In fact, the productivity of NL is significantly lower than the productivity of AV only for the pattern *list of* NE. Since this pattern corresponds to the largest corpus (see Table 1), its poor performance in acquisition has a strong impact on the overall productivity of NL. Avoiding this pattern would make NL more suitable for acquisition with a productivity of 23.2 (only 2 times lower than AV).

Ratios enumerations/lists. The ratios in the third lines of the tables correspond to the quotient of the number of candidates acquired by analyzing enumerations (`Pe` parser) to the number of candidates obtained from the analysis of lists (`Pl` parser). *Following* NE mainly yields NEs through the analysis of lists, probably because enumerations using coordinations are better introduced by *such as*. The outcome is more balanced for *list of* NE. It could be expected that this pat-

Table 1: Size of the corpora of HTML pages (in Mb) collected on the four patterns (1.a-d) through AltaVista (AV) and Northern Light (NL).

AV engine	*following* NE (AV)	*list of* NE (AV)	NE *such as* (AV)	*such* NE *as* (AV)
Largest corpus ORGANIZATIONS	6.1 *int. organizations*	6.4 *universities*	11.3 *int. organizations*	5.8 *int. organizations*
Largest corpus PERSON	5.8 *managers*	4.3 *journalists*	7.3 *politicians*	2.8 *musicians*
Largest corpus LOCATION	6.8 *countries*	4.9 *countries*	13.6 *states*	7.3 *states*
Total size	**85.9**	**64.9**	**150.4**	**66.3**

NL engine	*following* NE (NL)	*list of* NE (NL)	NE *such as* (NL)	*such* NE *as* (NL)
Largest corpus ORGANIZATIONS	10.0 *museums*	75.1 *int. agencies*	58.5 *holdings*	19.5 *universities*
Largest corpus PERSON	10.2 *actors*	60.0 *politicians*	44.1 *actors*	48.6 *authors*
Largest corpus LOCATION	23.0 *rivers*	61.2 *islands*	34.4 *rivers*	118.3 *states*
Total size	**172.8**	**1,306.9**	**652.7**	**458.1**

Table 2: Size of the number of candidate NEs acquired from the web-based corpora described in Table 1.

AV engine	*following* NE (AV)	*list of* NE (AV)	NE *such as* (AV)	*such* NE *as* (AV)
# candidates	4,747	3,112	5,738	3,579
Productivity	55.2	48.0	38.2	53.9
Ratio enum./list	0.28	0.83	12.5	43.74
Redundancy	2.12	2.15	1.77	1.69

NL engine	*following* NE (NL)	*list of* NE (NL)	NE *such as* (NL)	*such* NE *as* (NL)
# candidates	5,667	5,176	14,800	9,335
Productivity	32.8	4.0	22.7	20.4
Ratio enum./list	0.31	0.49	10.41	14.72
Redundancy	2.12	2.34	2.13	2.20

AV & NL	*following* NE	*list of* NE	NE *such as*	*such* NE *as*	**Total**
# candidates	8,673	7,380	18,005	10,566	**44,624**
Overlap	16.7%	11.0%	12.3%	18.2%	**15.0%**

tern tends to introduce only lists, but there are only 1.66 times more NEs obtained from lists than from enumerations through *list of* NE. The large number of NEs produced from enumerations after this pattern certainly relies on the combination of linguistics and formatting cues in the construction of meaning. The writer avoids using (the word) *list* when the text is followed by a (physical) list. Lastly, in all, 11 times more NEs are obtained from enumerations than from lists after the pattern NE *such as*, and 18 times more after *such* NE *as*. This shows that the linguistic pattern *such as* preferably introduces textual enumerations through coordinations (Hearst, 1998).

Redundancy. There are two main causes of redundancy in acquisition. A first cause is that the same NE can be acquired from several collections in the same corpus. Redundancy in the fourth lines of the tables is the ratio of duplicates among the yield of candidate NEs for each search engine and each query. This value is relatively stable whatever the search engine or the query pattern. On average, redundancy is 2.09: each candidate is acquired slightly more than two times. Acquisition through NL is slightly more redundant (2.18) than through AV (1.92). This difference is not significant since the number of NEs acquired through NL is twice as large as the number of NEs acquired through AV.

Overlap. Another cause of multiple acquisition is due to the concurrent exploitation of two search engines. If these engines were using similar techniques to retrieve documents, the overlap would be large. Since we have chosen two radically different modes of query (phrase vs. bag-of-word technique), the overlap—the ratio of the number common candidates to the number of total candidates—is low (15%). The two search engines seem to be complementary rather than competitive because they retrieve different sets of documents.

Precision of Acquisition

In all, 31,759 candidates are produced by postfiltering the acquisition from the corpora retrieved by the two search engines. A set of 504 candidates is randomly chosen for the purpose of evaluation. For each candidate, AV is queried with a phrase containing the string of the NE. The topmost 20 pages retrieved by AV are downloaded and then used for manual inspection in case of doubt about the actual status of the candidate. We assume that if a candidate is correct, an unambiguous reference with the expected type should be found at least in one of the topmost 20 pages.

Two levels of precision are measured:

1. A NE is correct if its full name is retrieved and if its fine-grained type (the 34 types given at the beginning of this section) is correct. The manual inspection of the 504 candidates indicates a precision of 62.8%.
2. A NE is correct if its full name is retrieved and if its MUC type (ORGANIZATION, PERSON, or LOCATION) is correct. In this case, the precision is 73.6%.

The errors can be classified into the following categories:

Wrong type Many errors in NE typing are due to an incorrect connection between a query pattern and a collection in a document. For instance, *Ashley Judd* is incorrectly reported as an athlete (she is an actress) from the occurrence

> *His clientele includes stars and athletes such as Ashley Judd (below) and Mats Sundin.*

The error is due to a partial analysis of the initializer (underlined above). Only *athletes* is seen as the hypernym while *stars* is also part of it. A correct analysis of the occurrence would have led to a type ambiguity. In this context, there is no clue for deciding whether *Ashley Judd* is a star or an athlete.

Other wrong types are due to polysemy. For instance, *HorseFlySwarm* is extracted from a list of actors in a page describing the commands and procedures for programming a video game. Here *actors* has the meaning of a virtual actor, a procedure in a programming environment, and not a movie star.

Incomplete Partial extraction of candidates is mainly due to parsing errors or to collections containing partial names of entities.

As an illustration of the second case, the author's name *Goffman* is drawn from the occurrence

> *Readings are drawn from the work of such authors as Laing,*

Szasz, Goffman, Sartre, Bateson, and Freud.

Since this enumeration does not contain the first names of the authors, it is not appropriate for an acquisition of unambiguous author's names.

Other names such as *Lucero* are ambiguous even though they are completely extracted because they correspond to a first name or to a name that is part of several other ones. They are also counted as errors since they will be responsible of spurious identifications in a name tagging task.

Over-complete Excessive extractions are due to parsing errors or to collections that contain words accompanying names that are incorrectly collected together with the name. For instance, *Director Lewis Burke Frumkes* is extracted as an author's name from a list in which the actual name *Lewis Burke Frumkes* is preceded by the title *Director*.

Miscellaneous Other types of errors do not show clear connection between the extracted sequence and a NE. They are mainly due to errors in the analysis of the web page.

These types of errors are distributed as follows: wrong type 25%, incomplete 24%, overcomplete 8% and miscellaneous 43%.

5 Refinement of the Types of NEs

So far, the type of the candidate NEs is provided by the NE hypernym given in (1.a-d). However, the initializer preceding the collection of NEs to be extracted can contain more information on the type of the following NEs. In fact the initializer fulfills four distinct functions:

1. introduces the presence and the proximity of the collection, *e.g. Here is*
2. describes the structure of the collection, *e.g. a list of*
3. gives the type of each item of the collection, *e.g. universities*
4. specifies the particular characteristics of each item. *e.g. universities in Vietnam*

The cues used by the harvester are elements which either introduce the collection (e.g. *the following*) or describe the structure (e.g. *a list of*). In initializers in general, these first 2 functions need not be expressed explicitly by lexical means, as the layout itself indicates the presence and type of the collection. Readers exploit the visual properties of written text to aid the construction of meaning (Péry-Woodley, 1998).

However it is necessary to be explicit when defining the items of the collection as this information is not available to the reader via structural properties. Initializers generally contain additional characteristics of the items which provide the differentia (underlined here):

This is a list of American companies with business interests in Latvia.

This example is the most explicit form an initializer can take as it contains a lexical element which corresponds to each of the four functions outlined above. It is fairly simple to extract the details of the items from initializers with this basic form, as the modification of the hypernym takes the form of a relative clause, a prepositional phrase or an adjectival phrase. A detailed grammar of this form of initializer is as shown in Figure 3.[5]

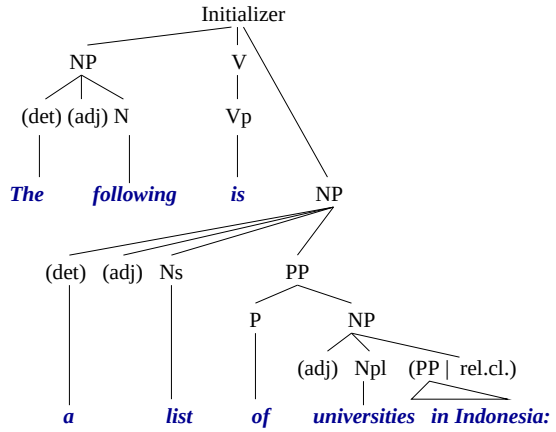

Figure 3: The structure of a basic initializer

We tag the collection by part of speech using the TreeTagger (Schmid, 1999). The elements which express the differentia are extracted by means of pattern matching: they are always the modifiers of the plural noun in the string, which is the hypernym of the items of the collection.

[5] PP = prepositional phrase, N_s = noun (singular), N_{pl} = noun (plural), V_p = verb in present tense, rel.cl. = relative clause.

Initializers containing the search string *such as* behave somewhat differently. They are syntactically incomplete, and the missing constituent is provided by each item of the collection (Virbel, 1985). These phrases vary considerably in structure and can require relatively complex syntactic rearrangement to extract the properties of the hypernym. We will not discuss these in more detail here.

One type of error in this system occurs when a paragraph containing the search string is followed by an unrelated list. For example the harvester recognizes

> *Ask the long list of American companies who have unsuccessfully marketed products in Japan.*

as an initializer when in fact it is not related to any collection. If it happened to be followed on the page by an collection of any kind the system would mistakenly collect the items as NEs of the type specified by the search string.

The cue *list of* is commonly used in discursive texts, so some filtering is required to identify collections which are not employed as initializers and to reduce the collection of erroneous items. Analyzing the syntactic forms has allowed us to construct a set of regular expressions which are used to eliminate *non-initializers* and disregard any items collected following them.

We have extracted 1813 potential initializers from the corpus of HTML pages collected via AV & NL for the query string *list of NE*. Using lexico-syntactic patterns in order to identify correct initializers, we have designed a shallow parser for filtering and analyzing the strings. This parser consists of 14 modules, 4 of which carry out pre-filtering to prepare and tag the corpus, and 10 of which carry out a fine-grained syntactic analysis, removing collections that do not function as initializers. After filtering, the corpus contains 520 collections. The process has a precision of 78% and a recall of 90%.

6 Conclusion

This study is another application that demonstrates the usability of the WWW as a resource for NLP (see, for instance, (Grefenstette, 1999) for an application of using WWW frequencies in selecting translations). It also confirms the interest of non-textual linguistic features, such as formatting markups, in NLP for structured documents such as Web pages. Further work on Web-based NE acquisition could take advantage of machine learning techniques as used for wrapper induction (Kushmerick et al., 1997).

References

E. Amitay. 1999. Anchors in context: A corpus analysis of web pages authoring conventions. In L. Pemberton and S. Shurville, editors, *Words on the Web - Computer Mediated Communication*, page 192. Intellect Books, UK.

C. Aone, N. Charocopos, and J. Gorlinski. 1997. An intelligent multilingual information browsing and retrieval system using Information Extraction. In *Proceedings, Fifth Conference on Applied Natural Language Processing (ANLP'97)*, pages 332–39, Washington, DC.

R. Basili, M.T. Pazienza, and P. Velardi. 1993. Acquisition of selectional patterns in sublanguages. *Machine Translation*, 8:175–201.

B. Boguraev and J. Pustejovsky, editors. 1996. *Corpus Processing for Lexical Acquisition*. MIT Press, Cambridge, MA.

F. Crimmins, A.F. Smeaton, T. Dkaki, and J Mothe. 1999. Tétrafusion: Information discovery on the internet. *IEEE Intelligent Systems and Their Applications*, 14(4):55–62.

B. Daille. 1996. Study and implementation of combined techniques for automatic extraction of terminology. In J.L. Klavans and P. Resnik, editors, *The Balancing Act*, pages 49–66. MIT Press, Cambridge, MA.

G. Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publisher, Boston, MA.

G. Grefenstette. 1999. The WWW as a resource for example-based MT tasks. In *Proc., ASLIB Translating and the Computer 21 Conference*, London.

M.A. Hearst. 1998. Automated discovery of WordNet relations. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

N. Kushmerick, D.S. Weld, and R. Doorenbos. 1997. Wrapper induction for information extraction. In *Proc., IJCAI'97*, pages 729–735, Nagoya.

M.-P. Péry-Woodley. 1998. Signalling in written text: a corpus based approach. In *Workshop on Discourse Relations and Discourse Markers at COLING-ALC'98*, pages 79–85.

H. Schmid. 1999. Improvements in part-of-speech tagging with an application to german. In S. Armstrong, K.W. Church, P. Isabelle, S. Manzi, E. Tzoukermann, and D. Yarowski, editors, *Natural Language Processing Using Very Large Corpora*. Kluwer, Dordrecht.

J. Virbel. 1985. Mise en forme des documents. *Cahiers de Grammaire*, 17.