Department of
Information Engineering
and Computer Science
DISI

UNIVERSITY
OF TRENTO - Italy

DISI - Via Sommarive 14 - 38123 Povo - Trento (Italy)
http://www.disi.unitn.it

LIGHTWEIGHT PARSING OF CLASSIFICATIONS

Aliaksandr Autayeu, Fausto Giunchiglia and Pierre Andrews

December 2010

Technical Report # DISI-10-068

Also: submitted to the "IJoDL Special Issue on ECDL 2010" - International Journal on Digital Libraries (IJoDL - http://www.springerlink.com/content/100475/)

Lightweight Parsing of Classifications

Aliaksandr Autayeu · **Fausto Giunchiglia** · **Pierre Andrews**

Abstract Understanding metadata written in natural language is a crucial requirement towards the successful automated integration of large scale, language-rich, classifications such as the ones used in digital libraries. In this article we analyze natural language labels used in such classifications by exploring their syntactic structure, and then we show how this structure can be used to detect patterns of language that can be processed by a lightweight parser whose average accuracy is 96.82%. This allows for a deep understanding of natural language metadata semantics. In particular we show how we improve the accuracy of the automatic translation of classifications into lightweight ontologies by almost 18% with respect to the previously used approach. The automatic translation is required by applications such as semantic matching, search and classification algorithms.

1 Introduction

The development of information technologies transformed the data drought into a data deluge, which seriously complicated the data management and information integration problems. This resulted in an increasing importance of metadata as a tool for allowing to manage data on a greater scale. The amount of existing attempts to solve the semantic heterogeneity problem shows its importance and reveals the variety of domains where it applies (see [7,6]). The state of the art algorithms try to solve the problem at the schema or metadata level [13] and their large-scale evaluations [16] show two important directions for improvement: a) increasing the background knowledge [14] and b) improving natural language understanding [28].

Digital library classifications extensively use natural language, both in structured and unstructured form, in particular in the labels of the nodes of the classification. These labels use a specific Natural Language (NL), different in structure from the normal textual domain of language, and the current natural language processing (NLP) technologies that are developed for the latter are not well suited for such the classif Natural Language Metadata (NLM). Thus, they require a domain adaptation to fit the specific constraints of the NLM structure. Moreover, the size of the current datasets [16], ranging from thousands to hundreds of thousands of labels (see Table 1), poses additional requirements on processing speed, as demonstrated by the LCSH and NALT alignment experiment from [15].

In general, the parsing of NLM has applications in many areas, in particular: a) in the *matching* of tree-like structures (such as Digital Libraries classifications or schemas) or lightweight ontologies [18], b) in the *Semantic Classification* of items of information into hierarchical classifications [19], and in c) *Semantic Search* [12]. All these *motivating applications* require the same steps of natural to formal language translation: a) recognize atomic (language-independent) concepts by mapping natural language tokens into senses from a controlled vocabulary, b) disambiguate the senses drawn from the controlled vocabulary and c) build complex concepts out of the atomic ones.

In this article we present the analysis of the natural language used in six classifications, which illustrate the use of NLM in classifications of information items in different domains. We show that the natural language used in these datasets is highly structured and can be accurately parsed with lightweight grammars. By using parsers based on these grammars, we allow for a deeper understanding of metadata semantics and improve the accuracy of the language to logic translation required by the semantic applications by almost 18% with respect to the previously used approach and without sacrificing performance.

This article is structured as follows. We introduce the classifications we study in Section 2. We follow with details on the processing steps required to understand the NLM,

E-mail: surname@disi.unitn.it
DISI, University of Trento, Italy

such as tokenization in Section 3 and Part-Of-Speech tagging in Section 4. We present the language structure analysis, based on the results of previous sections, in Section 5. Based on this analysis we have developed a set of grammars, which we describe in Section 6. We provide the evaluation of our approach in Section 7. We describe related work in Section 8. We conclude the article in Section 9.

2 Classifications

We have chosen to study a variety of classifications, which, we believe, illustrate well the use of natural language metadata. As a matter of fact, the classification we have cover different application domains, are created by paid professionals as well as unpaid contributors, and are of different size. The common theme among these metadata datasets is their use for classification, indexing and organization of items of information (e.g. documents).

LCSH[1] acronym stands for the "Library of Congress Subject Headings". It is a thesaurus of subject headings maintained by the U.S. Library of Congress for use in bibliographic records.

NALT[2] stands for "National Agricultural Library Thesaurus". NALT is a hierarchical vocabulary of agricultural and biological terms used extensively to aid indexing and retrieval of information within and outside of U.S. Department of Agriculture.

DMoz[3] or "Open Directory Project" is a web directory, collectively edited and maintained by a global community of volunteer editors.

Yahoo! Directory[4] is a "catalog of sites created by Yahoo! editors who visit and evaluate websites and then organize them into subject-based categories and subcategories".

eCl@ss[5] is an "international standard for the classification and description of products and services".

UNSPSC[6] stands for "United Nations Standard Products and Services Code". It is a "globally used classification hierarchy for products and services owned by the United Nations Development Programme (UNDP) and managed by GS1 US".

In the presented analysis, we use the hierarchical representation of all datasets where we select, for practical reasons, a random subset that is manually tokenized and annotated with part of speech tags using the PennTreeBank tag set [24]. The annotation was performed by a single expert annotator. Table 1 provides some key characteristics of our classifications.

[1] http://www.loc.gov/cds/lcsh.html
[2] http://agclass.nal.usda.gov/
[3] http://dmoz.org
[4] http://dir.yahoo.com/
[5] http://www.eclass-online.com/
[6] http://www.unspsc.org/

The datasets we use contain *subject headings*, *terms* and *category names*. Although called differently, all these items are written in natural language and are used for classification, search and indexing of items of information. Due to the common purpose and the many shared characteristics, we will use the word *label* to denote a single item of any of these datasets, be it a heading, a term or a category name.

We believe that these classifications are representative samples of Natural Language Metadata because their characteristics, such as topic coverage, editors and size, vary greatly. LCSH, DMoz and Yahoo have the widest coverage of all, eCl@ss and UNSPSC cover a wide variety of topics related to business and services, while NALT focuses on the agricultural domain. In addition, they differ in the competence of their editors as LCSH, NALT, eCl@ss and UNSPSC are edited by paid professionals, Yahoo is edited by editors, although anybody can suggest a site for inclusion, and DMoz is edited by a team of volunteers. Moreover, the datasets are of different size; Yahoo! directory is the largest one with 828 081 labels, while DMoz and LCSH follow being about two times smaller. NALT represents an even lighter "weight" category and eCl@ss and UNSPSC, by being about two times smaller, are the smallest ones as summarised in Table 1.

3 Tokenization

The results of the translation of NLM are employed to reason using concepts from a multilingual controlled vocabulary. These concepts are expressed in natural language as words. In most cases a concept expression in a language consists of a single token, separated from others by spaces. In other cases a concept might be expressed by several tokens, which might be separated by other punctuation signs, such as comma.

Tokenization is a preliminary step in almost any language processing; although a relatively simple task, in our domain of natural language metadata, standard tools meet several difficulties. These difficulties arise from various non-standard use of such punctuation elements as commas, round and square brackets, slashes, dashes, dots, ellipsis and semicolons: , () [] \ / : ... -. In addition, in several datasets we have noticed a non-standard use of punctuation, such as missing conventional space after a comma.

Consider the example "Hand tools (maint.,service)" from eCl@ss. This label uses a dot to abbreviate the word "maintenance" and is followed immediately by a comma without a conventional space afterwards, all of which is surrounded with round brackets. Such combinations are rare in normal texts and therefore the performance of standard tools, trained on such texts, degrades.

Table 1 Classification datasets' characteristics and example labels

Dataset	Labels	Sample Size	Unique Labels (%)	Levels	Label Length, NL tokens		Example Label
					Max	Avg	
LCSH	335 704	44 490	100.00	21	24	4.0	Concentration camp escapes
NALT	43 038	13 624	100.00	13	8	1.6	animal science
DMOZ	494 043	27 975	40.48	12	12	1.8	Arts and Entertainment
YAHOO	829 081	132 350	16.70	15	18	2.0	Political Actions Committees (PACs)
ECL@SS	14 431	3 591	94.51	4	31	4.2	Hand tools (maint.,service)
UNSPSC	19 779	5 154	100.00	4	19	3.5	Industrial Cleaning Services

3.1 Experimental Setup

A random subset of each dataset (see Table 1) is manually tokenized. We use the OpenNLP toolkit[7] to automatically annotate the full datasets. First, using the manually annotated subset of each dataset, we test the performance of the standard OpenNLP tokenization models, which are trained on the Wall Street Journal and Brown corpus data [22], which both contain long texts, mostly from newswire. Second, we train our own tokenization models and analyse their performance.

We performed a 10-fold cross-validation on each of our annotated samples (see Table 1) with the OpenNLP "standard" model, and also tested a *combined model* trained on the merged datasets. We report the results in the right part of Table 2 using precision per label (PPL) measure. Namely, we count the percentage of correctly tokenized labels. In each column we report the performance of different tokenizer models on a particular dataset; in the rows we report the performance of a model trained on a particular dataset, on the other datasets. Figures on the diagonal and for the combined model are obtained by a 10-fold cross validation.

3.2 Results and Analysis

The OpenNLP row reports the performance of the OpenNLP standard model. The last row is a combined model trained on a combination of the datasets available. Although in many cases the performance improvement is marginal, there are noticeable improvements in the cases of eCl@ss, Icon and LCSH. One can also notice that the model trained only on the LCSH dataset, which is particularly difficult, also outperforms the standard OpenNLP model.

The analysis of errors made by the tokenizer unveils that the main reason for this performance improvement is that punctuation is used in some short labels more intensively than in normal text. Therefore a retrained model grasps this difference better than the standard one.

[7] version 1.4.3, `http://opennlp.sourceforge.net/`

4 Part-Of-Speech Tagging

As discussed earlier, our main operational unit is a *concept*. There are hundreds of thousands of concepts expressed as words. Implementing a translation procedure based directly on words and on underlying concepts would be a complex task. A common approach is to use categories that words of natural language can be split into and thus simplify the task, achieving leverage over hundreds of thousands of words of natural language.

The categories above are called parts of speech (that is, Noun, Verb, etc.). In natural language processing it is common to assign a tag (part of speech tag) to each word, indicating the word's category. The translation routine then uses part of speech tags to handle categories of words, which are usually dozens, instead of words, which amount to hundreds of thousands. Part of speech (POS) tags also provide a significant amount of information about the language structure. This is why POS tagging is a fundamental step in language processing tasks such as parsing, clustering or classification. We, therefore, start our analysis with a look at the POS tags of our classifications.

4.1 Experimental Setup

As we did for the tokenization, a random subset of each dataset (see Table 1) is manually annotated by an expert with the PennTreeBank part-of-speech tag set [24]. We use the OpenNLP toolkit[8] to automatically annotate the full datasets. First, using the manually annotated subset of each dataset, we test the performance of the standard OpenNLP tokenization and tagging models, which are trained on the Wall Street Journal and Brown corpus data [22], which both contain long texts, mostly from newswire. Second, we train our own tokenization and tagging models and analyse their performance. We use the best performing models for the analysis of the full datasets presented in Sect. 5.

[8] version 1.4.3, `http://opennlp.sourceforge.net/`

Table 2 POS tagger and tokenizer performance, Precision Per Label, %

Model	POS Tagger						Tokenizer					
	DMoz	eCl@ss	LCSH	NALT	UNSPSC	Yahoo	DMoz	eCl@ss	LCSH	NALT	UNSPSC	Yahoo
DMoz	**93.98**	14.12	27.54	75.37	49.69	91.87	**99.95**	55.22	78.11	98.97	100.00	98.67
eCl@ss	48.80	**91.28**	28.60	28.73	69.65	62.11	99.73	**94.29**	97.70	99.97	99.98	99.45
LCSH	81.98	48.79	**91.38**	81.91	68.14	88.16	99.93	87.41	**99.79**	99.87	100.00	99.85
NALT	46.97	23.61	28.82	**96.42**	13.21	34.05	98.82	69.17	85.55	**100.00**	100.00	98.48
UNSPSC	57.07	45.08	22.76	31.03	**92.39**	75.46	97.09	47.98	43.63	98.80	**100.00**	96.76
Yahoo	89.54	15.20	34.84	75.04	45.91	**97.91**	99.90	55.69	88.47	99.12	100.00	**99.90**
OpenNLP	*49.89*	*19.02*	*27.26*	*40.55*	*33.20*	*47.44*	*99.86*	*79.39*	*95.57*	*99.96*	*100.00*	*99.77*
all-except	*91.59*	*58.40*	*53.25*	*84.77*	*76.19*	*94.77*						
path-cv	96.64	93.34	92.64	96.29	92.72	98.35						
combined	**99.10**	**99.69**	**99.24**	**99.74**	**99.40**	**99.68**	**99.95**	**94.26**	**99.51**	**100.00**	**100.00**	**99.90**

4.2 Results and Analysis

We report the results of our experiments in the left part of Table 2 where the columns report the dataset on which the experiments are run and the rows report the training model used. We report in bold the best performances. To indicate the percentage of correctly processed labels we report the precision per label. The figures on the diagonal and in the "path-cv" row are obtained by a 10-fold cross-validation. As a baseline, the "OpenNLP" row reports the performance of the standard OpenNLP tagging model.

The "path-cv" row reports the performance of the model where the labels appearing higher in the hierarchy were included in the context for training. Comparing the figures in bold with the figures in the "path-cv" row, we notice a performance increase of maximum 2.6%, with an average of 1.2%.

The "all-except" row is of particular interest, because it reports the performance of the model trained on all available datasets, except the one it will be tested on. For example, the model to be tested on DMoz data will include all datasets as training data, except DMoz itself. We can already notice a performance improvements compared to the standard OpenNLP model. The performance improvements are in a 25-47% range.

Finally, the "combined" row reports the performance of the model trained on a combination of datasets. This row demonstrates that the model we used to analyse the language structure has a high performance, this making the results of the analysis more reliable.

4.3 Observations and Discussion

We observe that NLM differs from the language used in normal texts. To assess whether NLM could be considered a separate language domain, we did cross-tests and took a closer look at the "all-except" row, comparing it with the "OpenNLP" one. In all the cases the performance is higher by a margin of 25%-47%. At the same time, the differences in model performance on different datasets are smaller than between the models. This performance evaluation confirms the difference between the NL used in metadata and in normal texts and it enables us to select the best applicable model for tagging unknown NLM.

Among the major reasons for such differences in performance, we see the lack of context in labels which is not an issue in long texts (see average label length in Table 1), the different capitalization rules between metadata and long texts, and the different use of commas. In addition, the POS tags distribution of labels is different from the one in normal texts as, for example, verbs are almost absent in NLM with, on average, 3.5 verbs (VB) per dataset, ranging from 0.0001% to 0.15% of all tokens of the dataset (see Fig. 2). Short labels mostly describe (sets of) objects and they do it by using proper and, often modified by adjectives, common nouns, more frequently than in normal text, where verbs constitute a larger portion of the words.

Furthermore, we performed an incremental training to evaluate whether our samples are large enough for the models to stabilize and found that the performances of our models stabilize around 96-98% precision per label on the size of our training samples. This shows that a larger manually annotated sample would not provide important accuracy improvements. Fig. 1 provide details for three larger datasets (LCSH, Yahoo and DMoz) in its left part and for three smaller datasets (eCl@ss, NALT and UNSPSC) in its right part.

5 Language Structure Analysis

To extract the meaning out of the labels, we should understand the structure of the language used to write such labels. We can then use the language structure for a more precise extraction of semantics out of labels by using a parser that is tuned to the specifics of their language.

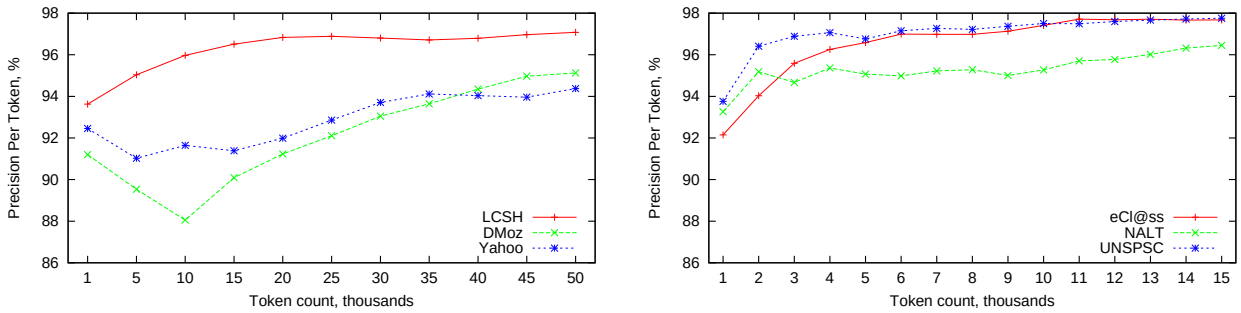

Fig. 1 POS tagger incremental training on LCSH, Yahoo and DMoz (left) and eCl@ss, NALT and UNSPSC (right)

The training of the part-of-speech (POS) tagger reported in the previous section enabled us to study the language structure of classification labels. We have analysed the labels' language structure by automatically POS tagging each dataset with the best performing model. We have found interesting repeating patterns, which we analyze and report here.

A *quantitative analysis* reveals some simple initial characteristics of the language used in natural language metadata. We use this and the additional data uncovered by quantitative analysis to perform a qualitative analysis. The *qualitative analysis* explores the use of commonly encountered syntactic features and language structures of labels. It also sheds some light on how syntactic features and language structures can be used to extract semantics out of labels in a more precise manner; in particular, by analysing POS tag patterns, we can derive grammars to generalize the parsing of the labels and simplify the translation to a formal language (see Sect. 6). This study also allows, by revealing the semantics of different pieces and elements of labels' pattern, to code "semantic actions" attached to appropriate grammar nodes in our lightweight parser to specialize the translation to the specific language constructions used in each dataset.

5.1 Quantitative Analysis

We studied such simple quantitative language characteristics as label length and POS tag distribution.

Our analysis of the label lengths (see Fig. 2) shows that the majority of labels is one to three tokens long. For example, more than half (50.83%) of all the DMOZ labels contain only one token. Two and three tokens labels represent 17.48% and 27.61%, respectively, while the longer labels only occur in less than 5% of the dataset. In comparison, the LCSH dataset tends to contain longer and more complex labels, with only 8.39% of them containing one token, 20.16% two tokens and about 10-14% for each of 3-, 4-, 5- and 6-token labels; the remaining 11.45% of labels contain more than 6 tokens. Differently to LCSH, almost all of the NALT labels are one and two tokens long. The amount of labels longer than 9 tokens in all datasets is less than 1% and we omit it from the graph.

Fig. 2 shows also the distribution of POS tags. We included all the tags that occur in more than 1% of all the tokens in any of the datasets analysed. Out of the 36 tags existing in the PennTreeBank's tagset [24], only 28 tags are used in the NLM datasets that we analysed. For comparison, we include POS tag distribution in normal text, represented by the Brown corpus [21].

We observe that all the datasets, except Yahoo, use less than 20 tags in total (see Table 3). Among the top ones are proper nouns (NNP, NNPS) and common nouns (NN, NNS), adjectives (JJ, JJR, JJS), conjunctions (CC), prepositions (IN) and punctuations (",", "(", ")"). Few verbs are also present, used as modifiers in the past form (VBD, max 0.0002%) and in the gerund form (VBG, max 0.08%).

5.2 Qualitative Analysis

The quantitative analysis of the previous section allowed us to discover several phenomena of the structure of natural language metadata. In the following paragraphs, we discuss the possible use of such structure in understanding the meaning of the labels.

5.2.1 Patterns

In each dataset we found specific repetitive combinations of POS tags, which we call "patterns" and illustrate in Table 8. Table 3 shows some characteristics of the language used in classifications with regard to these patterns. The column "90% coverage" shows the number of POS tag patterns required to cover at least 90% of the dataset.

The quantitative analysis uncovered a quite active comma use in LCSH. The comma is used in LCSH and also in eCl@ss to structure labels. LCSH labels are chunks of noun phrases, separated by commas, often in reverse order, such as in the label "Dramatists, Belgian" with the respective pat-

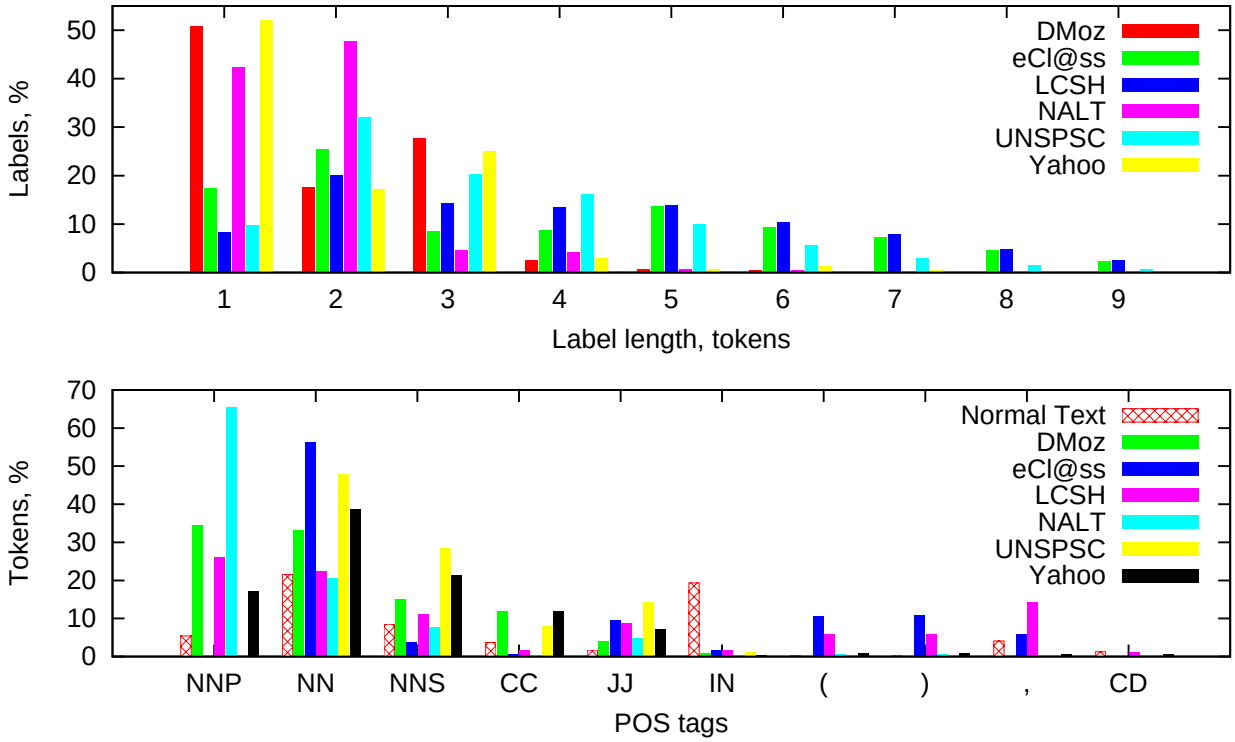

Fig. 2 Distributions of label lengths and POS tags

Table 3 Metadata language characteristics

Dataset	Tags	Patterns	90% Coverage	Top Pattern
LCSH	20	13 342	1 007	NNP NN
NALT	16	275	10	NNP NNP
DMOZ	18	975	9	NN
YAHOO	25	2 021	15	NN
ECL@SS	20	1 496	360	NN NN
UNSPSC	18	1 356	182	NN NNS

tern [NNS, JJ][9] covering 4 437 or 1.32% of all labels. There are also some naturally ordered examples, such as "Orogenic belts, Zambia" with the pattern [JJ NNS, NNP], which can be simplified into two noun phrase (NP) chunks [NP, NP] with independent structures. This pattern accounts for 1 500 or 0.45% of all labels.

A further examination of the LCSH patterns at the chunk level reveals that they form 44 groups. Each group consists of patterns where each piece has the same semantics. For example, the pattern [NNP NNP, NN CC NN, CD] of the label "United States, Politics and government, 1869-1877", when seen at the chunk level transforms into [geo, NP, time], where "geo" stands for a geographical proper name, "NP" stands for a noun phrase, and "time" stands for a time period. We can use the semantics of such chunks during the translation; For example, to leave out of the final translation result the tokens used for the disambiguation of other tokens, because once we have finished disambiguation they are no longer needed. We have identified the following chunk types:

- common noun phrase (NP): "International cooperation";
- event name (event): "Ashanti War";
- toponym (geo): "Tokyo";
- disambiguated toponym (geo-dis): "Tokyo (Japan)";
- time period (time): "1918-1945";
- noun phrase with disambiguation (NP-dis): "Contractions (Topology)";
- domain (domain): "in literature";
- personal name (name): "Constantine I";
- "wildcard": "Handbooks, manuals, etc.";
- "reversed" noun phrase: "Sculpture, Gothic".

Table 4 shows examples of five types of chunk pattern with their POS tag pattern and a sample label.

5.2.2 Content Features

A more thorough analysis of the labels' content reveals that, labels are almost exclusively noun phrases. Besides, DMOZ category names are clearly divided into the "proper" and "common" categories, which was noted in [28]. However, this is not the case for all datasets. We notice several interesting[10] groups of labels we have identified in DMoz:

[9] POS tags: NNS: plural noun, JJ: adjective, NNP: proper name, CD: cardinal number

[10] from the point of view of translation into logics

Table 4 5 LCSH Chunk Types with Examples

Chunk-Pattern	POS Tag Pattern and Label Example
event,geo,time	NNP NNP, NNP, NNP, CD Dock Strike, Manchester, England, 1951
event,time	NNP NNP, CD Turco-Montenegrin Wars, 1711-1714
event,time,geo	NNP NNP, CD, NNP World War, 1939-1945, Poland
event,time,NP	NNP NNP, CD, NNS Sino-Japanese War, 1894-1895, Causes
event,time,NP,geo	NNP NNP, CD, NNS, NNP Crimean War, 1853-1856, Campaigns, Romania

- **organization names**, containing a proper name and a noun, like "Union", "College", "Institute": like "Art Institute of Colorado", "Art Academy of Cincinnati", "Baptist College of Florida';
- patterns with commas, most of them **personal names**, like "Vives, Juan Luis", "Vries, Hugo de";
- patterns with other punctuation, most of them **movie or game titles**, like "Deception III – Dark Delusion", "Circus Maximus – Chariot Wars", "Ice Age – The Meltdown";
- structural patterns, or **facet indicators**, like "By Topic", "By Movement", "By Source of Exposure", "By Country of Service";
- "Series" patterns, like "Tetsuo Series", "Supercross Series", "DrumMania Series";
- **person names and movie titles in reverse**, like "Troup, Bobby", "Faculty, The";
- **"Based" labels**, like "Browser Based", "Fee Based", "Home Based";
- **organizations in reverse**, like "Education, Faculty of", "Engineering, College of";
- **personalized organization names**. These labels stand somewhat in between proper "proper names" and "common names", because on one hand they contain a proper name, but on the other hand they contain quite a lot of interpretable meaning, like in "Korea University of Technology and Education", or in "American Institute of Business and Economics";

Similar categories emerge in other datasets as well. Some labels combine into groups that encode special meaning or serve for structural purposes. We can use the POS tag patterns to recognize these special kinds of labels. For example, some structural labels resemble facet names or facet indicators.

The use of acronyms is another feature of labels we can notice. Although round brackets is a commonly used tool to introduce acronyms, this is different in natural language metadata. We illustrate such cases on the examples from UNSPSC, where many labels contain acronyms without any syntactic markers being used to indicate them. Such cases should be recognized and handled properly:

- an **acronym follows** tokens and their initial letters: "Light emitting diodes LEDs", "Central processing unit CPU processors";
- an **acronym contains initial letters of word components**: "Infrared IR sensors", "Polyvinyl Chloride PVC", "Polyethersulfone PES";
- an **acronym follows later**, not immediately after the abbreviated tokens: "Light rail vehicle transport LRV services";
- an **acronym does not correspond** to the letters or word components: "Acrylonitrile butadiene NBR";
- an **acronym precedes** abbreviated tokens: "VPN virtual private network managed network services".

These cases of acronym introduction should be distinguished from the cases where the acronym is simply used, like in "Programming for HTML" or "ERP or database applications programming services".

5.2.3 Syntax Tools

The quantitative analysis unveils a noticeable presence of punctuation and in particular of round brackets. This can be explained by their use as a disambiguation and specification tool, as illustrated by example labels "Watchung Mountains (N.J.)" and "aquariums (public)". Such a use of round brackets, if treated properly, helps in the formal language translation procedure. We illustrate the details of the use of punctuation by the examples of groups of labels from eCl@ss. We identify the following types of round brackets use:

- **specification**: "Laboratory app. (repair)", "epoxy resin (transparent)", "lithography (19th century)", "reducing flange (steel, alloyed)", "Screw (with head)";
- parts of **chemical slang**: "(E,E)-Potassium sorbate", "(S)-Malic acid". We note that chemical slang is regular and has precise semantics which can be parsed by a special grammar, however, exploiting this requires recognizing that these labels are indeed chemical and differentiating them from other labels;
- **repetition** of the broader topic from the above levels: "Seal, sealing material (packing material)", "Box (packing material)";
- **specification and repetition**: "Capsule (gelatine, packing material)", "Beaker (plastic, packing material)".

Often bracketed tokens repeat the label from the level above, but even in these cases the use is not consistent, although examples of the first of the following two kinds prevail. Compare:

- **Sub-topic (topic)**: "documentation (industrial compact computer)", "software (industrial compact computer)";

– **Topic (sub-topic)**: “industrial compact computer (accessories)”, “industrial compact computer (other)”

In the majority of cases, round brackets are to be found at the end of label. However, there are a few exceptions, such as: “Bottle (aluminum) larger than 1000 ml”, “Can (coex) up to 1000 ml”, “Cobalt (II) carbonate”, “Diethyl (trimethylsilyl) phosphite”, “Rhenium (IV) oxide”.

There are many cases were round brackets are repeated. Among these cases the following categories could be identified, with the first category prevailing:

– **specification**: “Reducing piece (high pressure) (non-ferrous metal)”, “T-piece (ready) (plast.)”, “Pipe (round) (non-ferrous metal)”, “Reducer (other) (glass)”;
– **specification and repetition**: “cutting grinder (electrical) (household appliance)”.

eCl@ss POS tag patterns containing commas constitute a significant (53.81%) portion of patterns. More than two thirds (68.07%) of patterns with commas contain commas present between tokens surrounded by brackets. We identify the following semantics of the pieces separated by commas within brackets:

– modifiers, specifying **different kinds of topic outside brackets**: “Threaded flange (iron, steel)”, “cross union (steel, alloyed)”;
– **modifier and repetition**: “Box (aluminum, packing material)”, “Carrier bag (paper, packing material)”, “Gun (steam, parts)”, “fork arm (industrial truck, parts)”;

Commas, used to separate pieces outside of round brackets, differ in their semantics too. We identify the following groups here, with the first group representing most of the cases:

– comma for enumeration of **largely independent pieces**: “Sound damper, pulsation damper”, “Machine, apparatus”, “Training, schooling”, “Cleansing material, cleaning material”;
– comma between **modifiers of a head noun**: “copying, printing line”, “Sparkling, dessert wine”;
– comma between **head noun and modifiers**: “Refrigeration, equipment”, “moistener, Finger Tip”, “Package insert, paper”, “window opener, electric”.

A few patterns (1.93%) also use a dash or a backward slash as a syntax tool, mostly to separate alternatives. However, while in some cases a dash or a slash indicate alternative, in others they separate a modifier or specifier. For example:

– **alternative**: “master clock / signal clock”, “account book / journal”, “Dewatering Machine - Expander/Expeller”, “softstarter/ AC-regulator”, “controller / card (PC)”;
– **modifier or specifier**: “Filter - Activated Carbon”, “Heat exchangers - reboiler”, “Sterilizer - Compression Still”.

6 Lightweight Parsing

From the analysis presented in the previous sections, we conclude that the parsing of labels in higher level structures can provide a better understanding of their semantics and thus to process them in a more meaningful notation for the computer. Following the motivating application a) from Sect. 1, we want to use the S-Match[11] [17] as implemented in [11] to align different classifications, such as in the experiment described in [15] and thus need a translation in a lightweight ontology, which would allow, for example, for the automatic integration of existing heterogeneous classifications.

Rule-based parsers use manually created linguistic rules to encode the syntactic structure of the language. These rules are then applied to the input text to produce parse trees. In long texts parsing, these have been disregarded because of two main disadvantages: they require a lot of manual work to produce linguistic rules and they have difficulties achieving a “broad coverage” and robustness to unseen data. To tackle these problems, state of the art statistical parsers, such as [2], infer grammar from an annotated corpus of text. However, this approach requires a large annotated corpus of text and a complicated process for tuning the model parameters. Moreover, producing a corpus annotated with parse trees is a much more costly and difficult operation than doing a basic annotation, such as POS tagging.

However, as we have seen in the previous section, in NLM, the language used is limited to (a combinations of) noun phrases. Hence, we need a limited coverage, which simplifies the construction of the rules. Therefore we use a simpler approach and manually construct a grammar for parsing. This requires having only an accurate POS tagging and some structural information of the language, which are provided by the analysis we described in the previous sections. We use a basic noun phrase grammar as a starting point for our grammars. Analyzing the POS tag patterns we modify this grammar to include the peculiarities of noun phrases as they are used in NLM, such as the use of commas and round brackets for disambiguation and specification (see examples in Sect. 5).

6.1 Grammars

We have developed a set of lightweight grammars for the datasets discussed in this paper. The grammars we created can be divided into two categories: “simple” ones with nine and ten rules (DMoz, eCl@ss and UNSPSC) and a “complex” ones with 15 and 17 rules (Yahoo, NALT and LCSH).

Fig. 3 shows two examples of the grammars we produced for the LCSH and UNSPSC datasets. For represent-

[11] see `http://semanticmatching.org`

Table 5 Grammar characteristics

Grammar	Rules	Coverage (%)		Parsing Mistakes (%)	
		Patterns	Labels	POS Tagger	Grammar
LCSH	17	92.96	**99.45**	49.59	47.94
NALT	15	59.27	**99.05**	80.35	13.30
DMOZ	9	90.95	**99.81**	85.98	11.01
YAHOO	15	65.31	**99.46**	70.90	20.50
ECL@SS	9	67.45	**92.70**	44.17	47.93
UNSPSC	10	70.58	**90.42**	25.01	65.70

ing the grammar rules we use the Backus-Naur Form (BNF). The LCSH one starts with a top production rule `Heading`, which encodes the fact that LCSH headings are made of chunks of noun phrases, which we call `FwdPhrase`. In turn, a `FwdPhrase` may contain two phrases `DisPhrase` with disambiguation elements as in the example above. The disambiguation element may be a proper noun phrase (`ProperDis`) or a common noun phrase (`NounDis`), surrounded by round brackets. `NounDis` is usually a period of time or a type of object, like "Fictitious character" in "Rumplemayer, Fenton (Fictitious character)" while `ProperDis` is usually a sequence of geographical named entities, like "Philadelphia, Pa." in "Whitemarsh Hall (Philadelphia, Pa.)". Fig. 5 contains the parse tree of this label, while Fig. 6 shows the parse tree of the example label from UNSPSC.

The core of the grammar is the `Phrase` rule, corresponding to the variations of noun phrases encountered in this dataset. It follows a normal noun phrase sequence of: a determiner followed by adjectives, then by nouns. Otherwise, it could be a noun(s) modified by a proper noun, or a sequence of foreign words.

6.2 Analysis and Discussion

A comparative analysis of the grammars of different classifications shows that they all share the nine base rules with some minor variations. Compare the rules 4-12 of LCSH with the rules 2-10 of UNSPSC in Fig. 3. These nine rules encode the basic noun phrase. Building on top of that, the grammars encode different syntactic tools used in different classifications for disambiguation and structural purposes. For example, in LCSH, a proper noun in a disambiguation element is often also disambiguated with its type, as "Mountain" in: "Nittany Mountain (Pa. : Mountain)".

A quantitative analysis shows the grammar coverage of the language, summarized in Table 5.

One can note that in all cases we have a high coverage of the dataset labels, more than 90% in all cases and more than 99% in four cases. If we look at the pattern coverage we notice a slightly different picture. For NALT, Yahoo, eCl@ss and UNSPSC, we have only 60% to 70% coverage of the patterns. This can be explained by Table 3 where, for instance, only around 1% of the patterns already cover 90% of the labels in NALT. This shows how a small amount of the labels uses a large variety of language construction while the majority of the NLM uses highly repetitive constructs.

Our analysis shows that the main reason for the lower coverage is a less regular use of language in these four classifications as compared to the other two classifications. We have analysed the mistakes done by the parser and found that they mostly fall into two major categories: POS tagger errors and linguistic rules limitations (see Table 5). This can be explained by the rule-based nature of our parser that makes it particularly sensitive to POS tagger errors. Other parser mistakes are due to the inconsistent (ungrammatical) or unusually complex labels, which could be seen as "outliers". For example, the "English language, Study and teaching (Elementary), Spanish, [German, etc.] speakers" label from LCSH contains both a disambiguation element "(Elementary)" and a "wildcard" construction "[German, etc.]".

Although very similar to one another, there are a few obstacles that need to be addressed before these grammars can be united into a single one. One of the most difficult of these obstacles is the semantically different use of round brackets: in most cases round brackets are used as a disambiguation tool, as illustrated by the examples mentioned above; however, we also found some examples where round brackets are used as a specification tool, as for instance in the label from eCl@ss: "epoxy resin (transparent)".

Due to these different semantics, these cases will almost certainly require different processing for a target application. For example, in translating metadata for semantic matching purposes [18], we need to translate the labels of a classification into a Description Logic formula to build up a lightweight ontology. In this application, the disambiguation element "(Pa. : Mountain)" of the label "Nittany Mountain (Pa. : Mountain)" can be used to choose a precise concept "Nittany Mountain" and the element itself is not included in the final formula, while in the specification case of "epoxy resin (transparent)", the specifier concept "transparent" should be included in the formula in a conjunction with the concept "epoxy resin" that is being specified.

Another obstacle is the different semantics of commas. Sometimes, a comma is used to sequence phrases. However, there are cases where the comma separates a modifier in a phrase, written in a "backward" manner, such as illustrated above with a label "Dramatists, Belgian". In long texts, these differences can be disambiguated by the context, which is almost always missing for NLM.

Despite these differences, our results show that simple and easily customizable grammars can be used to parse accurately most of the patterns found in the state of the art classifications, thus providing extra understanding of the NL without a loss in performance.

```
 1 Heading := FwdPhrase {"," FwdPhrase}
 2 FwdPhrase := DisPhrase {Conn} DisPhrase
 3 DisPhrase := Phrase {"("ProperDis | NounDis")"}
 4 Phrase := [DT] Adjs [Nouns] | [Proper] Nouns
           | Foreigns
 5 Adjs := Adj {[CC] Adj}
 6 Nouns := Noun {Noun}
 7 Conn := ConjConn | PrepConn
 8 Noun := NN [POS] | NNS [POS] | Period

 9 Adj := JJ | JJR
10 ConjConn := CC
11 PrepConn := IN | TO
12 Proper := NNP {NNP}
13 NounDis := CD | Phrase [":" Proper]
14 ProperDis := ProperSeq ":" Phrase
              | ProperSeq CC ProperSeq
15 Period := [TO] CD
16 ProperSeq := Proper ["," Proper]
17 Foreigns := FW {FW}
```

```
1 Label := Phrase {Conn (Phrase | PP$ Label)}


2 Phrase := Adjs [Nouns] | Nouns

3 Adjs := Adj {Adj}
4 Nouns := Noun {Noun}
5 Conn := ConjConn | PrepConn
6 Noun := NN [POS] | NNS [POS] | DT RB JJ
        | Proper
7 Adj := JJ | JJR | CD | VBG
8 ConjConn := CC | ,
9 PrepConn := IN | TO
10 Proper := NNP {NNP}
```

Fig. 3 LCSH (left) and UNSPSC (right) BNF production rules

Table 6 Example Label with Annotation

Annotation	Data			
tokens	Acupuncture	and	Chinese	Medicine
POS tags	NN	CC	JJ	NN
senses	n#699073	N/A	a#3048539	n#5964779
formula	n#699073 | a#3048539 & n#5964779			

Table 7 Evaluation Results Summary

Dataset	Labels	Accuracy (%)	Previously (%)	Improvement (%)
source	2 854	83.43	67.73	+15.70
target	6 628	81.05	65.89	+15.16

7 Evaluation

7.1 Setup and Methodology

We have evaluated the proposed solution using a synthetic approach. We have taken the large dataset used for evaluation of semantic matching [16], which is a technique used to identify semantically related information by establishing a set of correspondences, usually between two tree-like structures which are often denoted as "source" and "target". This dataset is a composition of three web directories: Google, Yahoo! and Looksmart. The "source" part contains 2 854 labels, while the "target" part contains 6 628 labels. We keep the dataset in two parts: "source", combined from Google and Looksmart directories, and "target", coming from Yahoo! directory, because these parts originate from different datasets, and this allows us to evaluate the performance on slightly different data. While containing parts of the Yahoo! directory and being from the same domain of natural language metadata, this dataset does not intersect with the ones we have used in our analysis discussed in Sect. 5 and for the training discussed in Sect. 4. Therefore it is appropriate to use it for evaluation purposes as it represents unseen data.

We have manually annotated this dataset with tokens, POS tags, assigned correct senses from WordNet [8] and, finally, created correct logical formula for every label. For example, for the label "Acupuncture and Chinese Medicine" we have the annotation displayed in Table 6.

Thus we have created a golden standard, which enables us to evaluate our approach. This dataset contains 47.86% of 1-token labels, 33.14%, 15.64% and 2.34% of 2-, 3- and 4-token labels, respectively. Longer labels constitute the remaining 1.02%. The average label length is 1.76 tokens, with the longest label being 8 tokens long. The most frequent POS tags are singular nouns (NN, 31.03%), plural nouns (NNS, 28.20%), proper nouns (NNP, 21.17%) and adjectives (JJ, 10.08%). An important POS, the coordinating conjunctions (CC) – which occupy a notable 6.58% – can introduce ambiguity in a label, which, in turn, might be carried into a formula. In total, 26 parts of speech are present, and except the ones already mentioned, other 21 parts of speech occupy the remaining 2.91%.

7.2 Results and Analysis

We summarize the evaluation results in Table 7. The column "Accuracy" contains the percentage of labels, for which our

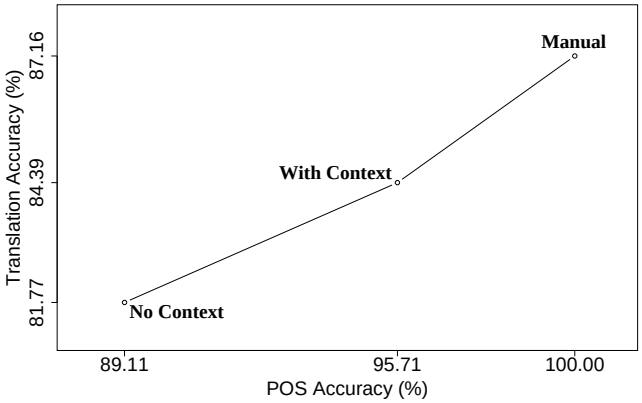

Fig. 4 Contribution of POS Accuracy to the Translation Accuracy

approach created correct formulas while the column "Previously" contains the accuracy of a previously used approach [17]. One can see that we have obtained a substantial improvement of approximately 15% over the previous results.

In Figure 4, we report the accuracy of the translation to description logic formulas, in comparison to the POS tagger performances. We report two different POS tagging models (see Sect. 4) on the combined "source+target" dataset:

- **No Context** that is the best *combined model*, and
- **With Context** that is the best combined model trained with contexts coming from classification paths of labels.

The best combined model reached 89.11% PPL on the combined "source+target" dataset. It compares well with the figures in the "all-except" row of Table 2 and shows that the model performs quite well on unseen data. We also tested the combined model trained with the context, and it reached 95.71% PPL. It compares well with the figures from the "path-cv" row of Table 2, also confirming that the model performs well on unseen data.

We can first observe an improvement of 6.6% in the POS tagging accuracy when using context, which stresses the importance of having a context. However, this only improves the translation accuracy by 2.62%. The improvement in POS tagging does not translate directly into a translation improvement, because of the other modules of a complete translation algorithm, such as the word sense disambiguation module, whose performance also influences the overall translation accuracy. Indeed, if we evaluate the translation with the manual POS tagging (*Manual* point in Figure 4), we observe that even with a "perfect" tagging, the translation accuracy does not improve much more. In comparison, a "perfect" tokenization (with a contextless POS tagging), improves the translation accuracy only by 0.02%.

To evaluate the influence of the tokenization and POS tagging preprocessing steps on the performance of the parser, we supplied the parser with correctly tokenized labels and it reached 81.79% precision. These 0.02% can give an estimation of the tokenizer contribution. Then we supplied the parser with the correct tags and it reached 87.16% precision. These 5.37% can give an estimation of the POS tagger contribution. Out of this experiment we see that improving the POS tagger can give us a 5.35% improvement, while the remaining 18.23% should be reached by improving other translation algorithm modules.

The analysis of mistakes showed that 19.87% (source) and 26.01% (target) of labels contained incorrectly recognized atomic concepts. For example, in the label "Diesel, Vin" two concepts "Diesel" and "Vin" were recognized, instead of the correct proper name: "Vin_Diesel". As another example consider the label "Early 20th Century", where the "previous" approach missed the concept "20th" because of too aggressive stopwords heuristics, while the proposed one recognized it. Vice versa, in the label "Review Hubs", instead of two concepts "Review" and "Hubs", only one wrong concept "Review_Hubs" was recognized. The cause of these mistakes is the POS tagger error because of the lack of context. Namely, the frequent misclassification which occurs between proper and common nouns. For these cases, further analysis of the erroneous formula does not make sense, because the atomic concepts are the basic building blocks of the formula, which should be recognized properly for the formula to be correct. For the rest, that is for the labels with correctly recognized atomic concepts, we found out that, in 49.54% (source) and 52.28% (target) of cases, the formula structure (that is, logical connectors or "bracketing") was recognized incorrectly. For example, in the label "Best & Worst Sites" the "&" sign is used as a conjunction, but was not recognized and this resulted in a wrong formula structure. The remaining half of the mistakes are word sense disambiguation mistakes of different kinds.

The approach we propose here, with more accurate NLP models and the parser based on the results of the language structure analysis, achieves the accuracy of 84.39% in the translation task. This is a 17.95% improvement over the state of the art translation approach [17] that reaches a 66.44% precision.

8 Related Work

The work available in the Semantic Web and Digital Libraries fields is often based on reasoning in a formal language (FL). However, users are accustomed to a NL and it is difficult for them to use a formal one. A number of approaches have been proposed to bridge the gap between formal languages and NL classifications.

Controlled languages (CLs), such as Attempto [9], have been proposed as an interface between NL and first-order logic. This, as well as a number of other proposals based on a CL approach [26,5], require the users to learn the rules and the semantics of a subset of English. Moreover, the users need to have some basic understanding of the first order

logic to provide a meaningful input. The difficulty of writing in a CL can be illustrated by the existence of editors, such as ECOLE [25], aiding the user in CL editing.

CLs are also used as an interface for ontology authoring [5, 1, 3]. The approach of [1] uses a small static grammar, dynamically extended with the elements of the ontology being edited or queried. Constraining the user even more, the approach of [3] enforces a one-to-one correspondence between the CL and FL. The authors in [5], following a practical experience, tailored their CL to the specific constructs and the errors of their users. Some of these and other CLs have been criticised [23] due to their domain dependance and genre limitations.

For querying purposes, [27] proposes an NL interface to ontologies by translating NL into SPARQL queries for a selected ontology. This approach is limited by the extent of the ontology with which the user interacts. In [4], the authors tackle limitations similar to the ones mentioned above and present the FREyA system, where they combine querying the underlying ontology with syntactic parsing. An interesting additional element of this approach is that authors involve a user by presenting clarification dialogs and using responses for training the system.

Another way to bridge the gap between formal languages and NLs is described in [10], where the authors propose to *manually* annotate web pages, rightfully admitting that their proposal introduces a "chicken and egg" problem. The approach described by [20] for automatically translating hierarchical classifications into OWL ontologies is more interesting, however, by considering the domain of products and services on the examples of eCl@ss and UNSPSC, the authors make some simplifying domain-specific assumptions, which makes it hard to generalise.

Differently from the approaches mentioned above, our work does not impose the requirement of having an ontology, the user is not required to learn a CL syntax, and we do not restrict our considerations to a specific domain. This article develops the theme of [28], improving it in several ways, such as extending the analysis to a wider sample of metadata and introducing a lightweight parser.

9 Conclusions

We have explored and analysed the natural language metadata represented in several large classifications. Our analysis shows that the natural language used in classifications is different from the one used in normal text and that language processing tools need an adaptation to perform well. We have shown that a standard part-of-speech (POS) tagger could be accurately trained on the specific language of the metadata and that we improve greatly its accuracy compared to the standard long texts models for tagging.

A large scale analysis of the use of POS tags showed that the metadata language is structured in a limited set of patterns that can be used to develop accurate (up to 99.81%) lightweight Backus-Naur form grammars. We can then use parsers based on these grammars to allow a deeper understanding of the metadata semantics. We also show that, for such tasks as translating classifications into lightweight ontologies for use in semantic matching it improves the accuracy of the translation by almost 18%.

10 Acknowledgments

This work has been partly supported by the INSEMTIVES project[12].

11 Appendix

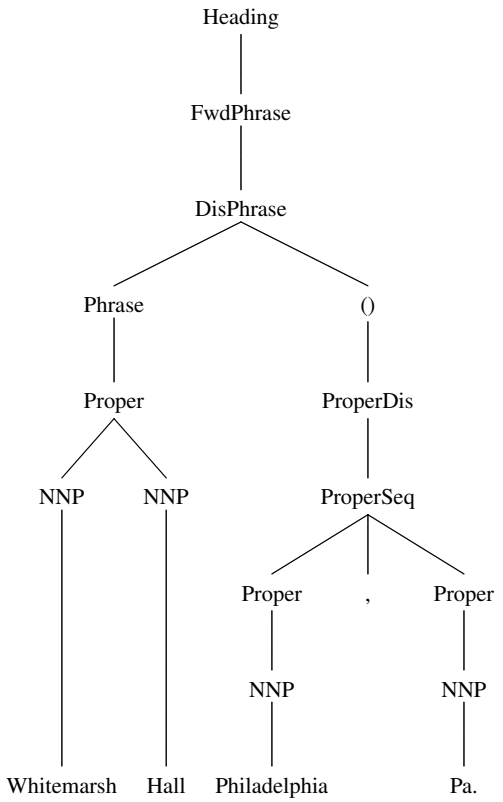

Fig. 5 Sample LCSH Label Parse Tree

References

1. Bernstein, A., Kaufmann, E.: GINO — a guided input natural language ontology editor. In: ISWC, pp. 144–157 (2006)

[12] FP7-231181, `http://www.insemtives.eu`

Table 8 Top 5 POS Tag Patterns with Examples by Dataset

Label Count	Share (%)	Pattern	Label Example
DMOZ			
104695	33.70	NN	Compensation
62625	20.16	NN CC NN	Pregnancy and Birth
44847	14.43	NNS	Sidecars
21854	7.03	NNS CC NNS	Magazines and Journals
13047	4.20	NN NNS	Restaurant Chains
ECL@SS			
2853	19.82	NN NN	Methyl benzoylformate
2457	17.07	NN	Acylase
583	4.05	NN NN (NN)	Laboratory app. (repair)
567	3.94	NN NN NN	Block heat exchanger
566	3.93	JJ NN	Exterior radiator
LCSH			
22192	6.61	NNP NN	Teach family
14444	4.30	NNP NNP (NNP)	Nishiki River (Japan)
13474	4.01	NNP	Myzocallis
11211	3.34	JJ NN	Negative staining
8771	2.61	NN NNS	Museum docents
NALT			
13356	31.03	NNP NNP	Rhode Island
12325	28.64	NNP	Diachros
3858	8.96	NN	thyroglobulin
2651	6.16	NN NN	milk allergy
2063	4.79	NNS	defoliants
UNSPSC			
3347	16.92	NN NNS	Sheet lifters
1662	8.40	NN NN NNS	Disc brake rotors
1511	7.64	NN NN	Play sand
1046	5.29	NNS	Levels
1009	5.10	JJ NNS	Brominated retardants
YAHOO			
211753	25.54	NN	Slowpitch
136156	16.42	NNS	Sidecars
84762	10.22	NN CC NN	Support and Assistance
52316	6.31	NNP	Hitwise
38395	4.63	JJ NN	High Jump

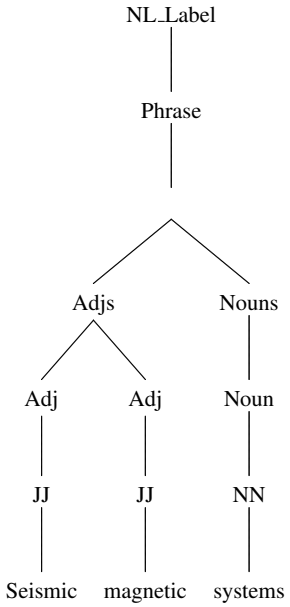

Fig. 6 Sample UNSPSC Label Parse Tree

2. Collins, M.: Head-driven statistical models for natural language parsing. Computational Linguistics **29**(4), 589–637 (2003)
3. Cregan, A., Schwitter, R., Meyer, T.: Sydney OWL syntax — towards a controlled natural language syntax for OWL 1.1. In: OWLED (2007)
4. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural language interfaces to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction. In: L. Aroyo, G. Antoniou, E. Hyvönen, A. ten Teije, H. Stuckenschmidt, L. Cabral, T. Tudorache (eds.) ESWC (1), *Lecture Notes in Computer Science*, vol. 6088, pp. 106–120. Springer (2010)
5. Denaux, R., Dimitrova, V., Cohn, A.G., Dolbear, C., Hart, G.: Rabbit to OWL: Ontology authoring with a CNL-based tool. In: CNL (2009)
6. Doan, A., Halevy, A.Y.: Semantic integration research in the database community: A brief survey. AI Magazine **26**, 83–94 (2005)
7. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer-Verlag (2007)
8. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. Language, Speech, and Communication. The MIT Press, Cambridge, MA (1998)
9. Fuchs, N.E., Kaljurand, K., Schneider, G.: Attempto controlled english meets the challenges of knowledge representation, reasoning, interoperability and user interfaces. In: FLAIRS Conference, pp. 664–669 (2006)
10. Fuchs, N.E., Schwitter, R.: Web-annotations for humans and machines. In: ESWC, pp. 458–472 (2007)
11. Giunchiglia, F., Autayeu, A., Pane, J.: S-Match: an open source framework for matching lightweight ontologies. Semantic Web Journal (2010)
12. Giunchiglia, F., Kharkevich, U., Zaihrayeu, I.: Concept search. In: ESWC (2009)
13. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: Semantic schema matching. In: Proceedings of CoopIS, pp. 347–365 (2005)
14. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: Discovering missing background knowledge in ontology matching. In: ECAI, pp. 382–386. IOS Press (2006)
15. Giunchiglia, F., Soergel, D., Maltese, V., Bertacco, A.: Mapping large-scale knowledge organization systems. In: ICSD (2009)
16. Giunchiglia, F., Yatskevich, M., Avesani, P., Shvaiko, P.: A large dataset for the evaluation of ontology matching systems. KERJ **24**, 137–157 (2008)
17. Giunchiglia, F., Yatskevich, M., Shvaiko, P.: Semantic matching: algorithms and implementation. In: JoDS, IX (2007)
18. Giunchiglia, F., Zaihrayeu, I.: Lightweight ontologies. In: EoDS, pp. 1613–1619 (2009)
19. Giunchiglia, F., Zaihrayeu, I., Kharkevich, U.: Formalizing the get-specific document classification algorithm. In: ECDL, pp. 26–37 (2007)
20. Hepp, M., de Bruijn, J.: GenTax: A generic methodology for deriving OWL and RDF-S ontologies from hierarchical classifications, thesauri, and inconsistent taxonomies. In: ESWC, pp. 129–144 (2007)
21. Kucera, H., Francis, W.N., Carroll, J.B.: Computational Analysis of Present Day American English. Brown University Press (1967)
22. Morton, T.: Using semantic relations to improve information retrieval. Ph.D. thesis, University of Pennsylvania (2005)

23. Pool, J.: Can controlled languages scale to the web? In: CLAW at AMTA (2006)
24. Santorini, B.: Part-of-speech tagging guidelines for the Penn Treebank Project. Tech. rep., University of Pennsylvania (1990). (3rd revision, 2nd printing)
25. Schwitter, R., Ljungberg, A., Hood, D.: ECOLE — a look-ahead editor for a controlled language. In: EAMT-CLAW, pp. 141–150 (2003)
26. Schwitter, R., Tilbrook, M.: Lets talk in description logic via controlled natural language. In: LENLS (2006)
27. Wang, C., Xiong, M., Zhou, Q., Yu, Y.: Panto: A portable natural language interface to ontologies. In: ESWC, pp. 473–487 (2007)
28. Zaihrayeu, I., Sun, L., Giunchiglia, F., Pan, W., Ju, Q., Chi, M., Huang, X.: From web directories to ontologies: Natural language processing challenges. In: ISWC/ASWC, pp. 623–636 (2007)