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 CLASSICATIONS INTO LIGHTWEIGHT ONTOLOGIES

Aliaksandr Autayeu, Fausto Giunchiglia, and Pierre Andrews

March 2010

Technical Report # DISI-10-025

Also: submitted to the European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2010)

Lightweight Parsing of Classifications into Lightweight Ontologies

Aliaksandr Autayeu, Fausto Giunchiglia, and Pierre Andrews

DISI, University of Trento, Italy

Abstract. Understanding metadata written in natural language is a premise to successful automated integration of large scale, language-rich, classifications such as the ones used in digital libraries. We analyze the natural language labels within classification by exploring their syntactic structure, we then show how this structure can be used to detect patterns of language that can be processed by a lightweight parser with an average accuracy of 96.82%. This allows for a deeper understanding of natural language metadata semantics, which we show can improve by almost 18% the accuracy of the automatic translation of classifications into lightweight ontologies required by semantic matching, search and classification algorithms.

1 Introduction

The development of information technologies turned the data drought into a data deluge, which seriously complicates data management and information integration problems. This leads to an increasing importance of metadata as a tool allowing the management of 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 [1, 2]). The state of the art algorithms try to solve the problem at the schema or metadata level [3] and their large-scale evaluations [4] show two important directions for improvement: a) increasing the background knowledge [5] and b) improving natural language understanding [6].

Digital library classifications extensively use natural language, both in structured and unstructured form. Natural language metadata (NLM) uses a specific Natural Language (NL), different in its structure from the normal textual domain of language, and the current NL processing (NLP) technologies that are developed for the latter are not well suited for NLM. Thus, they require a domain adaptation to fit the specific constraints of the NLM structure. Moreover, the size of the current datasets [4], 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 [7].

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 [8], b) in the *Semantic Classification*

Table 1. Classification datasets' characteristics

Dataset	Labels	Sample Size	Unique Labels (%)	Levels	Label Length, NL tokens	
					Max	Avg
LCSH	335 704	44 490	100.00	21	24	4.0
NALT	43 038	13 624	100.00	13	8	1.6
DMoz	494 043	27 975	40.48	12	12	1.8
Yahoo	829 081	132 350	16.70	15	18	2.0
eCl@ss	14 431	3 591	94.51	4	31	4.2
UNSPSC	19 779	5 154	100.00	4	19	3.5

of items of information into hierarchical classifications [9], and in c) *Semantic Search* [10]. All these *motivating applications* require the same steps of natural to formal language translation: a) recognize atomic (language-independent) concepts by mapping NL 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.

We present the analysis of the NL used in six classifications: **LCSH** [1] (for "Library of Congress Subject Headings"), **NALT** [2] (for "National Agricultural Library Thesaurus"), **DMoz** [3] (for Open Directory Project), **Yahoo! Directory** [4] (a "catalog of sites created by Yahoo! editors"), **eCl@ss** [5] (a classification of products and services), **UNSPSC** [6] (for "United Nations Standard Products and Services Code"), which all illustrate the use of NLM in classifications of information items in different domains. Note that, these datasets contain *subject headings*, *terms* and *category names*, which are all written in NL and which we hereafter refer to as *label(s)*. Table 1 provides some key characteristics of our classifications. We show that the NL used in these datasets is highly structured (see Sections 3 and 4) and can be accurately parsed with lightweight grammars (see Sect. 5). 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% (see Sect. 6) without sacrificing performance.

2 State of the Art

The work available in the semantic web and Digital Libraries is often based on reasoning in a formal language (FL). However, users are accustomed to a NL

[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/

and it is difficult for them to use a formal one. A number of approaches has been proposed to bridge the gap between formal languages and NL classifications.

Controlled languages (CLs), such as Attempto [11], 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 [12, 13], require users to learn the rules and the semantics of a subset of English. Moreover, 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 [14], aiding the user in CL editing.

CLs are also used as an interface for ontology authoring [13, 15, 16]. The approach of [15] 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 [16] enforces a one-to-one correspondence between the CL and FL. The authors in [13], 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 critiqued [17] due to their domain and genre limitations.

For querying purposes, [18] proposes an NL interface to the 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. Another way to bridge the gap between formal languages and NLs is described in [19], 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 [6], improving it in several ways, such as extending the analysis to a wider sample of metadata and introducing a lightweight parser.

3 Part-Of-Speech Tagging

Parts of speech (POS) tags provide a significant amount of information about the language structure. The POS tagging is a fundamental step in language processing tasks such as parsing, clustering or classification. This is why we start our analysis with a look at the POS tags of our classifications.

A random subset of each dataset (see Table 1) is manually tokenized and annotated by an expert with the PennTreeBank part-of-speech tag set [21]. 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 and tagging models, which are trained on the

[7] `http://opennlp.sourceforge.net/`

Table 2. POS tagger performance, Precision Per Label, %

Model	DMoz	eCl@ss	LCSH	NALT	UNSPSC	Yahoo
DMoz	**93.98**	14.12	27.54	75.37	49.69	91.87
eCl@ss	48.80	**91.28**	28.60	28.73	69.65	62.11
LCSH	81.98	48.79	**91.38**	81.91	68.14	88.16
NALT	46.97	23.61	28.82	**96.42**	13.21	34.05
UNSPSC	57.07	45.08	22.76	31.03	**92.39**	75.46
Yahoo	89.54	15.20	34.84	75.04	45.91	**97.91**
OpenNLP	*49.89*	*19.02*	*27.26*	*40.55*	*33.20*	*47.44*
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**

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 the next section. In addition, 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.

We report the results of our experiments in Table 2 where the columns report the dataset on which the experiments are run and the rows the training model used. As baseline, the "OpenNLP" row reports the performance of the standard OpenNLP tagging model. The "all-except" row reports the performance of the model trained on all datasets except the one it will be tested on to show robustness across datasets and on unseen data. 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. Finally, the "combined" row reports the performance of the model trained on a combination of datasets. The figures on the diagonal and in the "path-cv" row are obtained by a 10-fold cross-validation. We report in bold the best performances. To indicate the percentage of correctly processed labels we report the precision per label.

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.

As 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 for 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. 1).

4 Language Structure Analysis

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

For instance, the comma is widely used in LCSH and eCl@ss to structure the labels. LCSH labels are chunks of noun phrases, separated by commas, often in reverse order, such as in the label "Dramatists, Belgian" with the pattern [NNS, JJ][8] 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.

We studied some other language characteristics as well, such as label length and POS tag distribution, with which, in addition to the patterns, we can derive grammars to generalize the parsing of the labels and simplify the translation to a formal language (see Sect. 5). This study also allows, by revealing the semantics of different pieces and elements of labels' pattern, to code "semantic actions" attached to the appropriate grammar nodes in our lightweight parser to specialize the translation to the specific language used in the dataset.

Our analysis of the label lengths (see Fig. 1) 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 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. 1 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 from the PennTreeBank's tagset [21], 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 [23].

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

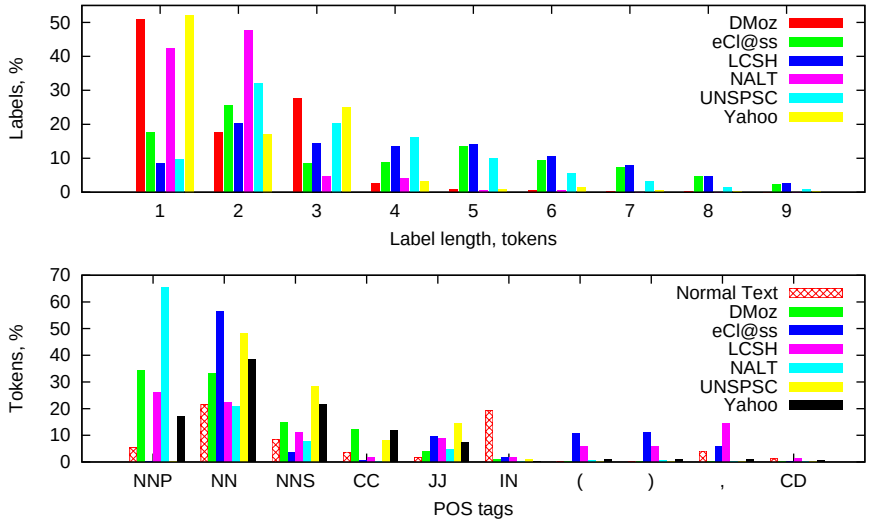

Fig. 1. Distributions of label lengths and POS tags

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 (“,” and “(”, “)”). A small amount of verbs is present, used as modifiers in the past form (VBD, max 0.0002%) and in the gerund form (VBG, max 0.08%).

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

In each dataset we found specific repetitive combinations of POS tags (referred to as patterns). Table 3 shows some characteristics of the language used in classifications with regard to these patterns. The column “90% coverage” shows count of POS tag patterns required to cover at least 90% of the dataset.

A qualitative analysis reveals more details. For example, labels are almost exclusively noun phrases. DMOZ category names are clearly divided into the

"proper" and "common" categories, which was noted in [6]. However, this is not the case for all datasets. Also a noticeable presence of round brackets is explained by their use as a disambiguation and specification tool, as illustrated by the labels "Watchung Mountains (N.J.)" and "aquariums (public)", which, if treated properly, helps in the formal language translation procedure.

When studying the LCSH patterns at a chunk level (using commas as separators) we can identify 44 groups of chunk-patterns, where many chunks bear clear semantics. For example, the pattern [NNP NNP, NN CC NN, CD] of the label "United States, Politics and government, 1869-1877", when seen at a 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.

5 Lightweight Parsing

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 motivating application a) from Sect. 1, we want to use the S-Match algorithm [24] to align different classifications, such as in the experiment described in [7] and thus need a translation in a lightweight ontology, which. This allows, 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 [25], 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. 4).

We have developed a set of lightweight grammars for the datasets discussed in this paper. The grammars we constructed 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). Table 4 provides details about the grammar coverage.

Table 4. Grammar characteristics

Grammar	Rules	Coverage (%)		Parsing Mistakes (%)	
		Patterns	Labels	POS Tagger	Grammar Rules
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

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 4). 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.]".

Fig. 2 shows two examples out of the grammars we produced for the LCSH and UNSPSC datasets. We use Backus-Naur form (BNF) for representing the grammar rules. The LCSH one starts with a top production rule `Heading`, which encodes the fact that LCSH headings are built 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.)".

```
 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. 2. LCSH (left) and UNSPSC (right) BNF production rules

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. Alternatively, it could be a noun(s) modified by a proper noun, or a sequence of foreign words.

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. 2. These nine rules encode the basic noun phrase. Building on top of that, the grammars encode the differences in syntactic rules used in different classifications for disambiguation and structural purposes. For example, in LCSH, a proper noun in a disambiguation element is often further disambiguated with its type, as "Mountain" in: "Nittany Mountain (Pa. : Mountain)".

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 [8], 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 a concept "epoxy resin" that is being specified.

Another obstacle is the different semantics of commas. Sometimes, a comma is used to indicate a sequence of 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.

6 Evaluation

We have evaluated our approach in a semantic matching application with the dataset from [4] that contains 9 482 labels from a variety of web directories. We have manually annotated all this dataset with tokens, POS tagging information and assigned a correct logical formula to every label. For example, we have annotated the label "Religion and Spirituality" with the POS tags "NN CC NN" and the formula "n#5871157 | n#4566344", where n#ID point to WordNet synsets for "religion" and "spirituality", respectively, and | stands for logical disjunction, which was lexically expressed with "and". The average label length is of 1.76 tokens, with the longest label being of 8 tokens. 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%).

In Fig. 3, we report the accuracy of the translation to description logic formulas, in comparison to the POS tagger performances. We consider the translation to be correct if the resulting formula is logically equivalent to the formula in the manual annotation. We report two different POS tagging models (see Sect. 3): **No Context** that corresponds to the best combined model, **With Context** that is the best combined model trained with a context coming from the classification path of the labels.

We can first observe an improvement of 6.6% in the POS tagging accuracy when using the context, which stresses the importance of such context. However, this only improves the translation accuracy by 2.62%. The improvement in POS tagging does not translate directly into a translation improvement, due to the

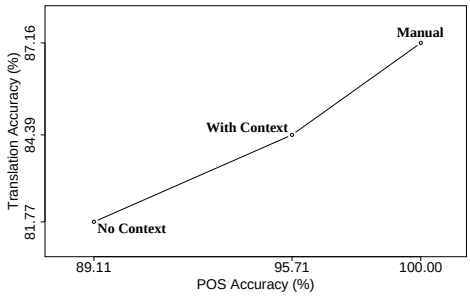

Fig. 3. Contribution of POS accuracy to the translation accuracy

other modules of the pipeline, 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 Fig. 3), 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%.

The approach we propose in this paper, with more accurate NLP models and the language structure analysis, achieves an accuracy of 84.39% in this application domain. This is a 17.95% improvement over the state of the art translation approach from [24] that reaches a 66.44% precision.

Analysing the errors, we observe that incorrect recognition of atomic concepts accounts for 22.94% of wrongly translated labels. In the remaining 77.06% of wrongly translated labels the errors are split into two groups: 79.29% due to incorrectly disambiguated senses and 20.71% due to incorrectly recognized formula structure. This suggests directions for further improvements of the approach.

7 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%.

References

1. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer-Verlag (2007)
2. Doan, A., Halevy, A.Y.: Semantic integration research in the database community: A brief survey. AI Magazine **26** (2005) 83–94
3. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: Semantic schema matching. In: Proceedings of CoopIS. (2005) 347–365
4. Giunchiglia, F., Yatskevich, M., Avesani, P., Shvaiko, P.: A large dataset for the evaluation of ontology matching systems. KERJ **24** (2008) 137–157
5. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: Discovering missing background knowledge in ontology matching. In: ECAI, IOS Press (2006) 382–386
6. 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. (2007) 623–636
7. Giunchiglia, F., Soergel, D., Maltese, V., Bertacco, A.: Mapping large-scale knowledge organization systems. In: ICSD. (2009)
8. Giunchiglia, F., Zaihrayeu, I.: Lightweight ontologies. In: EoDS. (2009) 1613–1619
9. Giunchiglia, F., Zaihrayeu, I., Kharkevich, U.: Formalizing the get-specific document classification algorithm. In: ECDL. (2007) 26–37
10. Giunchiglia, F., Kharkevich, U., Zaihrayeu, I.: Concept search. In: ESWC. (2009)
11. Fuchs, N.E., Kaljurand, K., Schneider, G.: Attempto controlled english meets the challenges of knowledge representation, reasoning, interoperability and user interfaces. In: FLAIRS Conference. (2006) 664–669
12. Schwitter, R., Tilbrook, M.: Lets talk in description logic via controlled natural language. In: LENLS. (2006)
13. Denaux, R., Dimitrova, V., Cohn, A.G., Dolbear, C., Hart, G.: Rabbit to OWL: Ontology authoring with a CNL-based tool. In: CNL. (2009)
14. Schwitter, R., Ljungberg, A., Hood, D.: ECOLE — a look-ahead editor for a controlled language. In: EAMT-CLAW. (2003) 141–150
15. Bernstein, A., Kaufmann, E.: GINO — a guided input natural language ontology editor. In: ISWC. (2006) 144–157
16. Cregan, A., Schwitter, R., Meyer, T.: Sydney OWL syntax — towards a controlled natural language syntax for OWL 1.1. In: OWLED. (2007)
17. Pool, J.: Can controlled languages scale to the web? In: CLAW at AMTA. (2006)
18. Wang, C., Xiong, M., Zhou, Q., Yu, Y.: Panto: A portable natural language interface to ontologies. In: ESWC. (2007) 473–487
19. Fuchs, N.E., Schwitter, R.: Web-annotations for humans and machines. In: ESWC. (2007) 458–472
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. (2007) 129–144
21. Santorini, B.: Part-of-speech tagging guidelines for the Penn Treebank Project. Technical report, University of Pennsylvania (1990) (3rd revision, 2nd printing).
22. Morton, T.: Using Semantic Relations to Improve Information Retrieval. PhD thesis, University of Pennsylvania (2005)
23. Kucera, H., Francis, W.N., Carroll, J.B.: Computational Analysis of Present Day American English. Brown University Press (1967)
24. Giunchiglia, F., Yatskevich, M., Shvaiko, P.: Semantic matching: algorithms and implementation. In: JoDS, IX. (2007)
25. Collins, M.: Head-driven statistical models for natural language parsing. Computational Linguistics **29**(4) (2003) 589–637