University of Bergen

Master Thesis

Metadata made easy - Adding metadata by means of natural language

Author:
Snorre Magnus DavÃŸen

Supervisor:
Richard Elling Moe

in the

Semantic and Social Information Systems
Department of Information Science and Media Studies

September 2013

“It depends upon what the meaning of the word ’is’ is"

– Bill Clinton

UNIVERSITY OF BERGEN

Abstract

Faculty of Social Sciences

Department of Information Science and Media Studies

Masters degree

Metadata made easy - Adding metadata by means of natural language

by Snorre Magnus DavÃÿen

Abstract comes here!

Acknowledgements

Acknowledgements!

Contents

List of Figures

List of Tables

List of Listings

Nomenclature

CSS	Cascading Style Sheets
HTML	HyperText Markup Language
IRI	Internationalized Resource Identifier
OWL	Web Ontology Language
RDF/XML	An XML representation of RDF
RDFa	Resource Description Framework in attributes
RDF	Resource Description Framework
SUMO	Suggested upper merged ontology
SUO	Standard Upper Ontology
Synset	A set of words with similar meanings.
URL	Uniform Resource Locator

Chapter 1

Introduction

Introduce the background for the master thesis work. What is information retrieval, why is it used. Shortly talk about clustering and classification and how it is used to help make order in large document collections. What kind of problems might clustering solve?

Introduce the Klimauken project, and the compact trie clustering algorithm.

1.1 Motivation

Academic motivations:

The Compact Trie Clustering algorithm has some benefits (being computationally fast, supporting phrases), but performs poorly on data sets which have not been pre-filtered by say a search engine. How can this master thesis work contribute to the performance/-effectivity of the algorithm. Why is this a interesting area to research.

Other motivations:

Information retrieval an interesting field. Genetic algorithms an interesting means to the optimization of large parameter sets. Computing fitness with genetic algorithm a challenging and exiting task. The opportunity to learn a new programming language (Python).

1.2 Research question

The main research question of this thesis is:

"Insert research question here"

Explanation of research question here...

A list of subsidiary goals here:

- Goal 1
- Goal 2

Why is this an interesting research question, and how does it relate to the motivation.

1.3 Target audience

Researchers and users of the suffix tree/compact trie clustering algorithm in general, and the research group working with Klimauken and other language processing research in particular.

Limitations made to the research work should be listed here.

Chapter 2

Theory

Introduction to chapter

2.1 Clustering and information retrieval

A general introduction to text document clustering should be introduced here.

2.1.1 Suffix Trees and Suffix Tree Clustering

What is a suffix tree? How is one built?

2.1.2 Compact Trie Clustering

What is the compact trie clustering algorithm...

2.1.3 Performance measures

What kind of performance measures are used for clustering...

2.1.4 Available corpora

Which corpora are used in clustering and/or classification research? Which ones are suited to clustering? Which ones are used in this master thesis research? Explain scope...

2.2 Genetic Algorithms

General overview of a genetic algorithm here.

2.3 Related Work

Introduce related research work in this chapter.

Chapter 3

Methodology

An introduction to the chapter.

3.1 Experimental Evaluation

Will rewrite and use the corresponding section from the project proposal.

3.1.1 Evaluation Measures

Provide info about the two forms of evaluation measures used (one from Improved suffix tree clustering article, and one used by Richard). How do they work, what are the differences...

3.2 Corpora

Write a bit about the corpora used in the research

3.3 Experimental Research

Rewrite and use corresponding section from project proposal. Try to flesh out the methodology a bit (how did I perform the testing, what where the hypotheses etc). Talk about experimental constraints and data used.

Chapter 4

Development and Testing

Chapter introduction. Describe the development process in some detail. I.e. The learning phase (learning Python and familiarizing myself with the algorithm). The second stage (modifying the algorithm slightly, refactoring). Third stage (implementing a genetic algorithm with which to test different parameter sets). Fourth stage (making a distributed version of the algorithm to make testing of larger parameter sets feasible/possible). Fifth stage (final touches on the algorithm, implementing support for converting other corpora to the snippet file format). (Sixth stage) Storing results in a database and developing a method of extracting statistical numbers. Overview of system?

Describe how the algorithm and parameters where tested. I.e. how data and results were gathered. Write about different testing stages. First stage (testing individual parameter options to determine usable value ranges. Are results correct, why?). Testing of non-distributed genetic algorithm (does the small test-case give indication of viability?). Testing distributed algorithm (larger parameter sets tested show better results?). Final testing (testing hypotheses, include two corpora?).

4.1 Stages of development

This section will outline the main sequence in which work on MaDaME was done. The overview of the iterations will explain when the initial development of the different parts were started to give an short explanation of the development process.

4.1.1 Learning stage

Familiarize with algorithm ...

4.1.2 Modification of algorithm

Modified so and so ...

4.1.3 Genetic algorithm

Explain how and why it was developed as it was ...

4.1.4 Distribution of genetic algorithm

Why distribute? How? Possible problems...

4.1.5 Results storage and corpus processing

Storing results, tracking top chromosomes over generations, extracting averages for graphs etc.

4.1.6 Overview of the completed system

Give a short overview of the system ...

4.2 Testing

Give overview of testing process

4.2.1 Value Range Tests

4.2.2 Genetic Algorithm Test

4.2.3 Distributed Genetic Algorithm Test

4.2.4 Final testing

Chapter 5

Analysis and Discussion

This chapter will contain an analysis of the different parts of the system and discuss their appropriateness in solving the problems tied to the research question. The chapter will start with comparing the mapping algorithms to see if one of them performs better then the other. It will then try to answer the questions of whether MaDaME was able to generate metadata of equal or better quality than that which is already available, and if the system is capable of adding the metadata without changing the way the Web page is displayed. It will end with an overview of the usage data the system has received at this point and see how it corresponds with the expected use of the system.

5.1 Comparing the algorithms

To compare the two algorithms a list of English nouns was generated. This list was sent through the lexitags server to get synsets that corresponded to the meanings of each word. Both these lists were preprocessed to remove duplicates and to format them as JavaScript objects [1]. The final list of synsets contained 4350 unique synsets. A short script was written that was used to run the synsets through the best-fit algorithms, and to write a report of the results. For the schema.org version of the test the script wrote the average depth of the mapped type, as well as the total number of times the two algorithms had the same and different mappings. The depth was calculated as the distance from the root node in the tree, i.e. schema:Thing itself had a depth of 0, schema:Person which

[1] `https://github.com/EivindEE/Madame/tree/master/testing`

inherits directly from schema:Thing has a depth of 1 and so on. For the SUMO version the depth of each type was unavailable, so the test results from this test only show the agreement between the algorithms. The full results from the tests can be found at the URL `https://github.com/EivindEE/Master-thesis/tree/master/AlgComparison`.

As one can read from the numbers in table 5.1 the results from the SUMO test display no difference between the two algorithms when mapping from WordNet to SUMO. The two algorithms return identical mappings in 100% of the test cases. This indicates that there must have been a mapping either directly from the synset, or from the direct hypernym of the synset for each of the 4350 synsets in the test. This makes it hard to say anything relevant about the algorithms from these results.

	Schema.org	SUMO
Total number of synsets tested	4350	4350
Number of identical mappings	3262 (75%)	4350 (100%)
Number of different mappings	1088 (25%)	0 (0%)
No result found	598 (13.7%)	0 (0%)

Table 5.1: The testing results

The schema.org results are more interesting. The two algorithms still perform fairly equally. Reading from table 5.1 we can see that the algorithms are equal in 75% of the cases, but we can examine the 25% that are different and see which perform better in those cases. We can also see that the algorithms were unable to find a mapping in 13.7% of the cases. These cases cannot tell us much about which algorithm should be preferred, but could be a good starting point for finding parts of the WordNet to schema.org mapping which could be enhanced.

The prediction made beforehand was that the hypernyms first approach would have fewer incorrect mappings, but would give results at a more shallow depth. The last prediction was the easiest to test, as we know the schema.org hierarchy and can calculate the depth of each type. For each mapping the script running the test registered the depth of the type in the schema.org hierarchy. These depths were averaged over the total number of mappings made.

As seen in table 5.2 both algorithms map fairly high in the hierarchy. The hypernyms first approach maps to a type at level 0.69 on average when considering all the synsets, or to a type at level 0.72 when ignoring the cases where the two algorithms gave the same result. As predicted the hypernym then siblings approach does a little better, though not

	Hypernyms First	Hypernym then siblings
For all mappings		
Avg. depth total	0.688506	0.804138
For different mappings		
Avg. depth different	0.721507	1.183823
Mappings to schema:Thing	449	334
Mappings to schema:Intangible	481	81
For the 250 examined mappings		
Correct mappings	235	67
Correct mapping rate	94%	26.8%
Unclear	10	27
Unclear rate	4%	20.8%
Errors	5	156
Error rate	2%	62.4%

Table 5.2: Comparison of the mapping algorithms

much, mapping to types at level 0.8, or 1.2 when excluding identical mappings. Looking at the data it is obvious that the hyponyms first approach much more frequently leads to mappings to schema:Thing and schema:Intangible. The hypernym first algorithm maps to schema:Thing 449, and schema:Intagible 481 times, while hypernyms then siblings maps to schema:Thing 334 and schema:Intangible 81 times. Neither schema:Thing nor schema:Intangible are very interesting mappings in the ontology. As described in section **??**, schema:Thing is the most general category, meaning that every concept belongs to this category. On the other hand, schema:Intangible is described as "a utility class that serves as the umbrella for a number of 'intangible' things", and does not have any special properties in the ontology.

These results were a bit disappointing. One should however keep in mind the fact that schema.org is not a general ontology. As mentioned in section **??** the schema.org ontology is geared towards things that are relevant to search engines. The synsets that were used in this test covered a wide variety of topics. The top-level categories of schema.org were shown in figure **??** on page **??** When looking at the results it is the necessary to keep in mind that synsets that do not belong to any of those 10 categories cannot have a better mapping than schema:Thing in the ontology.

5.1.1 Correctness of the algorithms

To find which algorithm performed best it was decided to check the results where the algorithms gave different mappings. In the cases where the mappings were the same, the two algorithms obviously performed equally well since they mapped to the same concept. The results from when the two algorithms gave different mappings they were checked manually to judge their correctness. The process consisted of looking up each synset that was mapped, and the types it had been mapped to and check if the two corresponded. The results were divided into three categories. If it was clear that the synset and type corresponded, they were marked as correct. If it was clear that they did not correspond they were marked as incorrect. There were also some cases where it was unclear whether or not a mapping was correct.

It was decided that it was necessary to include a category for unclear mappings to highlight the fact that some of the categories are fuzzy and require some more documentation, or that they might entail things than seem unnatural. One of the instances were it was unclear if a mapping should be judged to be correct was for the mapping of dairy#n#1, "a farm where dairy products are produced", which maps to schema:FoodEstablishment. Schema:FoodEstablishment is described as "[a] food-related business", which a dairy most certainly is. The sub typing of schema:FoodEstablishment however seems to indicate otherwise. The sub types of schema:FoodEstablishment are:

- Bakery
- BarOrPub
- Brewery
- CafeOrCoffeeShop
- FastFoodRestaurant
- IceCreamShop
- Restaurant
- Winery

This seems to indicate an establishment where private customers come to purchase goods, making a more industrial venue seem out of place. Another schema.org term that provided some difficulty was schema:Place, which has the description "[e]ntities that have a somewhat fixed, physical extension". Again the sub types seem to indicate that it should be used for geographical sections. From the description of the type it is unclear if it can be used to describe things like borders and edges of things. This would depend on what the thing should be fixed with regard to.

Since the manual inspection of the mappings was a time consuming process I decided to only inspect 250 mappings, and see if the results of checking these would be sufficient to say anything about the algorithms. Mappings to schema:Thing and schema:Intangible were excluded as one could normally argue reasonably for these.

A higher error rate in the hypernym then sibling algorithm had been predicted, but the difference in the error rate between the algorithms was much larger than anticipated. Again pointing to table 5.2 we can see that the hypernyms first algorithm made correct mappings in 94% of the test cases, while giving incorrect mappings in 2.0% test cases and questionable mappings in 4% of the cases. The hypernym then sibling algorithm on the other hand gave a correct mapping in only 26.8% of the cases checked. It gave incorrect mappings in 62.4% and unclear mappings in 20.8% of the cases. The fact that it gave correct mappings at a rate of close to 25% of the instances where the results were different was very surprising. This high degree of incorrect mappings indicates that using sibling synsets as the basis for mapping in unfruitful. The causes of the incorrect mappings will be examined in the next section, to see if they are caused by weaknesses in the algorithms, or in WordNet or the mapping from WordNet to schema.org.

5.1.2 Analyzing the sources of error

The fact that the hypernyms first algorithm gave incorrect mappings at all is a bit alarming. As described in section **??** about hyper- and hyponymy, hypernymy should be a transitive "type of" relation. Each hypernym should then be a more general type of the synset provided and not break the semantics of the synset. The mappings from WordNet to schema.org indicate that the semantic content of the synset are equal to the semantic content of the schema.org type mapped to. Since both hypernymy and mappings should preserve the semantics of the synsets, the mappings should be correct. The fact that the hypernyms first algorithm gives incorrect mappings indicate that either the integrity of the hypernym relation is broken, or that the mappings are incorrect.

All the cases where the hypernyms first approach was deemed to have incorrect mappings were instances where the synset mapped to schema:Quantity via their hypernym measure#n#2 {measure, quantity, amount}. We can see in figure 5.1 how the different synsets are related to measure#n#2.

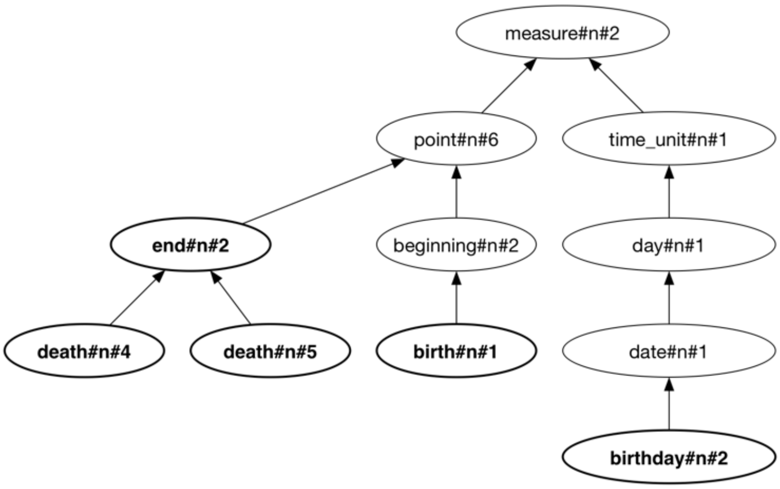


Figure 5.1: How the synsets with incorrect mappings are related to measure#n#2

In schema.org the type schema:Quantity is described as "[q]uantities such as distance, time, mass, weight". The synset measure#n#2 is described as "how much there is or how many there are of something that you can quantify". The mapping between these appears reasonable and do not seem to be the cause of error.

What makes the mappings seem unreasonable is that the synsets that were mapped incorrectly represent events that happen at a single instance in time. The problem in the chains seem to be that one goes from a synset which represents something that happened at a single instance in time, to a synset which describes a quantity of things. Using the "is-a" test, it does not seem reasonable that an english user would say that a birth is a type of quantity. The cause of the errors then is an inconsistency within the hypernym relation in WordNet.

I have reported these errors to the maintainers of WordNet, and they will be fixed in a future public release of WordNet (C. Fellbaum, personal communication, May 13, 2013).

The results where the hypernym then siblings algorithm mapped incorrectly were also analyzed. The distribution of incorrect mappings can be seen in figure 5.2 (page 36).

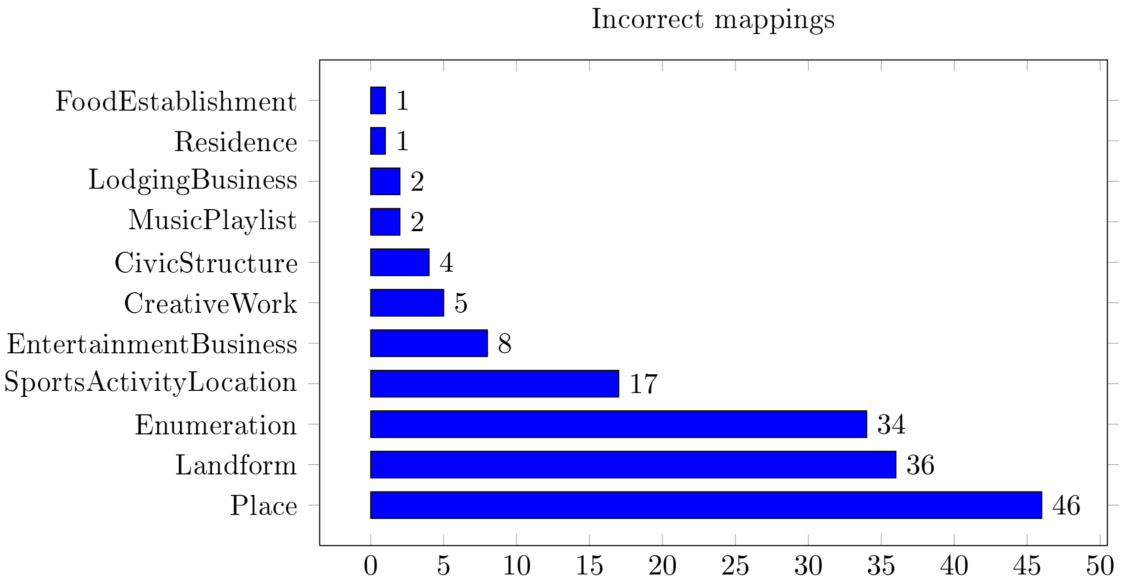


Figure 5.2: Distribution of incorrect mappings in hypernym then siblings algorithm

The two most common incorrect mappings were picked as objects of further study. I went through all the cases of incorrect mapping to schema:Place. This examination showed that in 45 of the 46 cases the mapping to schema:Place came as a result of the synset being a hyponym of whole#n#2, which has the sibling geological_formation#n#1 which again has a mapping to schema:Landform, a sub type of schema:Place. The last instance is the mapping from attention#n#3 which has the sibling tourist_attraction#n#1 mapping to schema:TouristAttraction.

For the mappings to schema:Landform all of the incorrect mappings came as a result of the synsets being hyponyms of part#n#3. That synset has the hypernym thing#n#12, which has the hyponym body_of_water#n#1. Body_of_water#n#1 maps to schema:BodyOfWater, which again is a sub type of schema:Landform. Common for all these mappings is that the hypernyms are reasonable. There is no reason to believe that these are cases where the hypernym chain breaks the semantics of the synset. The mappings from WordNet to schema.org also seem to be sound.

The analysis seems to exclude both the hypernym relation and the mappings from WordNet to schema.org as causes for the incorrect mappings. That leaves the possibility that using the siblings caused the error. Using the mappings from the siblings were wrong in these cases, as the sibling had some other semantic value than the synset for which the algorithm tried to find a mapping.

There were instances of siblings both of the original synset and of the hypernyms, so it does not appear that closeness to the synset needs to have any influence on whether or not the sibling gives an accurate mapping for the synset.

5.1.3 Choosing an algorithm

From the previous discussion and from the results of the testing it seemed clear that the hypernyms first algorithm is the dominant strategy for mapping synsets to schema.org, and that there is no difference between the two when mapping to SUMO. The algorithms performed very differently when mapping to the different ontologies.

It might be that an increase in the number of mappings would change the results. One could then try to refine the hypernym then siblings algorithm to analyze the mappings of its siblings, and try to find some more sophisticated way of choosing which mapping to select.

It would also be interesting to try mapping to a more general ontology than schema.org. As mentioned in section **??** schema.org is a small ontology with only 577 types, and with a bias towards commercial interests. It might be that an ontology that was larger, and which had a more balanced approach to the world would result in better results for the algorithm.

As it stands it is clear that the hypernyms first algorithm out performed the hypernym then siblings algorithm, and is the one that will be used in the artefact. The fact that its mappings were incorrect or questionable in 6% of the instances analyzed is unfortunate. The cause of the errors were however found to be outside the system that has been developed for this thesis, and they have been reported. This gives us reason to hope that the error rate of the system will go down as the tools it depends on are improved. It also suggests that some error reporting mechanism should be created to give feedback to the maintainers when incorrect mappings or hypernym relations occur.

5.2 Testing against existing markup

One of the success criteria was that the artefact should be able to mark up HTML as well or better than what is on the Web now. To test if MaDaME managed this I used Web pages that were already marked up with schema.org markup, and tried to generate similar markup using MaDaME before comparing the results to see if the metadata created by the tool was of a similar quality.

5.2.1 Method for comparing the results

The method that was used to mark up and compare the sites followed a simple process. Web pages were picked from a list of sites using the schema.org ontology for metadata that is available on GitHub [2].

The markup of the Web pages that were picked was examined to find which things that were marked up, and which schema.org types they were marked up as. This process was selected to make sure that it would be possible to compare the results.

The pages were imported into the artefact, and the markup would be recreated by selecting text, and if necessary supply keywords to disambiguate the content. For some sections supplying keywords were required as the text did not contain keywords that signaled the meaning of the content. There were several reasons for this. In some cases the keyword was displayed on the Web page as an image, so instead of having the term "review" in a header or in text it appeared as part of an image on the page. It could also be that the individual words on the page did not represent the concept. In other cases the difficulty was tied to mapping to functional concepts. For example is being for sale not an inherent property of thing it self, making mappings to products difficult using descriptions of the thing.

Metadata was added as faithfully as possible to make comparison simpler. When the markup process was complete, the Web pages would be exported and the resulting Web page would be analyzed. The original was used as a gold standard which the marked up pages could be compared to.

[2] `https://github.com/LawrenceWoodman/mida/wiki/Sites-Using-Microdata`

The analysis of the metadata would be performed by using Google's structured data testing tool [3], and W3s RDFa 1.1 distiller and parser[4]. The structured data testing tool shows the metadata that is read and extracted by Google. It was used to check if the metadata available to Google was the same. The tool was used on both the original page and the page marked up by the artefact so that one could judge if the extracted metadata was the same.

The RDFa distiller could not be used to compare the documents as it is created to distill RDFa not microdata, which was used to include metadata on the pages tested against. The RDFa distiller parses the Web page and extracts the RDFa, displaying it as some RDF format. It was used to see that the markup that was created was valid RDFa and that it could be translated to RDF.

The pages that were marked up using this process were:

- A restaurant review from the Telegraph
- A tour operators customer feedback page
- A tourist agency home page
- The home page of a marketing company
- A movie review sites review of a film

The full HTML of the Web pages before and after they were marked up using MaDaME can be found at `https://github.com/EivindEE/Master-thesis/tree/master/WildTesting`.

Most of the metadata on the Web pages that were marked up was metadata about larger sections of the page. The artefact is targeted towards disambiguating single words as described in section 4.2. When marking up the text this meant one had to decide whether to markup the same section of the text as used in the original page, or if one should select a single word in the section describing its content. Should one for example select the header "Review", or the entire review itself? Choosing the first option would give the metadata a different structure, while selecting the second option would require the user to provide a keyword describing the topic of the section. There is little practical difference

[3] `http://www.google.com/webmasters/tools/richsnippets`

[4] `http://www.w3.org/2012/pyRdfa/`

between the two strategies. Selecting a section might make it easier for humans looking at the markup to find out which part of the page the type refers to, but MaDaME does not use the content of the tag to add information so there is no difference in the semantic meaning of the metadata that is added. Neither should there be any difference in how the Web pages are displayed after the metadata is added. Marking up an larger section does however increase the amount of places where an error could occur. A combination of these two strategies was tried when marking up the pages.

5.2.2 An issue with the testing tool

An issue that was discovered when analyzing the resulting markup with the structured data tool was that it returned error messages saying that some of the elements that had been marked up were missing required properties. When analyzing a schema:Product without a name value the error message reads:

```
Warning: Missing required field "name (fn)".
Warning: Incomplete rdfa with schema.org.
```

The documentation on the schema.org homepage does not give any indication that schema types have required properties. The initial reaction to this was annoyance that these fields were required by RDFa but not by microdata. Closer inspection of the unmodified Web pages however revealed that when using the microdata format these types were ignored without warning. It is positive that the RDFa created by the artefact gives an error message instead of failing silently, but it is not good that the testing tool describes a field as required when this is not mentioned in the documentation. The fact that the validator requires the properties does not in itself mean that the markup is incorrect, or that the syntax is wrong. The validator is targeted towards the generation of rich snippets, and it might be that the warning is intended to warn that the amount of metadata is insufficient to generate rich snippets. The warning that required fields are missing might then refer to the fact that they are required to create a snippet, not required for the metadata to be valid. The metadata generated in these instances turned out to be of equal quality as that of the original documents, but also showed that the tool could do more to promote attributes that are required by search engines.

5.2.3 Comparison of the results

During testing it was realized that there are some schema.org types that the system was not able to reach. In particular it was discovered that the system does not have a way to reach the aggregated schema.org types. At the time of writing the aggregate types in schema.org are schema:AggregateRating and schema:AggregateOffer. The type schema:AggregateRating is meant to represent the average rating that a rated object has received. The type schema:AggregateOffer on the other hand represents a collection of offers for a given product. The difficulty posed by both these types is that what separates them from their super types schema:Rating and schema:Offer is that they represent the plurality of the super type. The system uses WordNet to represent and disambiguate words, and as its basis for mapping natural language to ontologies. WordNet is a dictionary type system, which does not separate between the different grammatical numbers of a given word, as they all represent the same concept. This could be an issue for the system as a whole since it means that the language used as an intermediary between natural language and the ontologies does not capture all of the complexity of the ontologies the system is supposed to map to.

The schema.org types that were used on the Web pages were (the number of times used in parenthesis):

- Review (11)
- Product (11)
- ImageObject (11)
- TravelAgency (3)
- Article (3)
- Rating (1)
- Movie (1)
- Person (1)
- People (3 - non-existing type)

The types that the Web pages mapped to were so high level that they all had natural direct mappings, meaning that the best-fit algorithm did not have to be used. Adding the properties to these were simple, as the tool provided a list of the properties that the type was able to have.

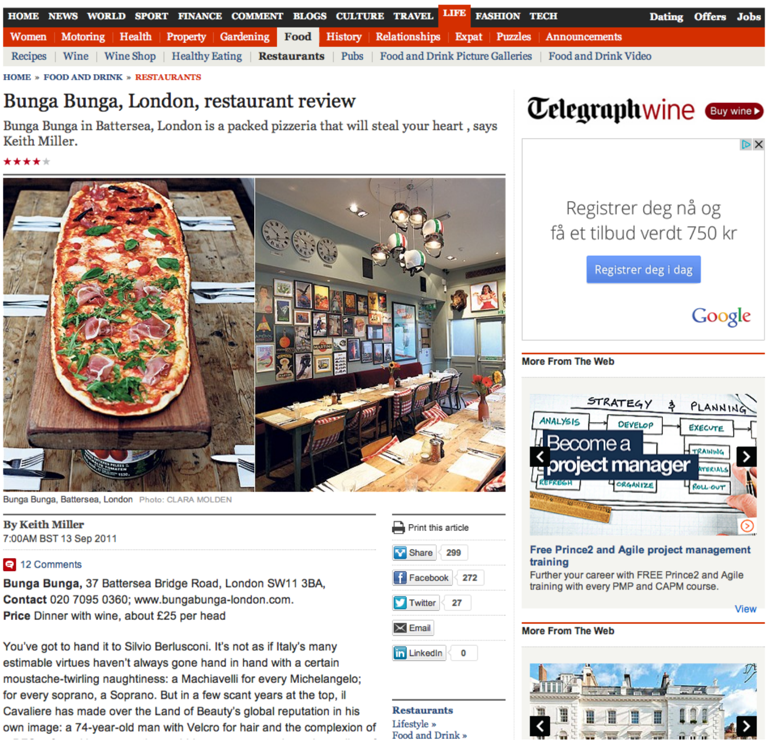
It was found that one of the pages used illegal markup. The page with the movie review it was found that the author of the HTML had illegal types and properties. The movie review page referred to people as schema:People, while the correct type in schema.org is schema:Person. The page used this pseudo-type to refer to multiple people by providing a name property to the type for each person. This usage in not possible in MaDaME as it does not allow multiple properties. The Web page also used the property name "publishdate" while the correct name of the property is "datePublished".

Mistakes like these are simple to make since it is natural for humans to think that people and person, or publishing date and date published are two ways of saying the same thing. This is however the ambiguity that we want to remove for computers by using metadata. An advantage of using a tool like MaDaME is that one can be sure not to use incorrect types or properties such as this. The tool will only allow types and properties that exist in the schemas that are included.

MaDaME was able to create mappings to all the types that were used on the Web pages. It was also able to add all the properties that the types had. In addition it was able to correct incorrect markup. These results show that the tool is able to create metadata of equal quality as that which is present on the Web now.

5.3 Browser rendering

One of the criteria for the artefact was that it should leave the visual representation of the system unchanged. The system is meant to let users add metadata to existing Web pages, and should not modify their appearance. I did a comparison of the Web pages before and after adding metadata. An example of the telegraph Web site before adding metadata can be seen in figure 5.3, and after in 5.4. The side margins and the header of the Web site has been cropped out in the images to make it easier to see the content of the sites.



HOME NEWS WORLD SPORT FINANCE COMMENT BLOGS CULTURE TRAVEL LIFE FASHION TECH Dating Offers Jobs
Women | Motoring | Health | Property | Gardening | Food | History | Relationships | Expat | Puzzles | Announcements
Recipes | Wine | Wine Shop | Healthy Eating | Restaurants | Pubs | Food and Drink Picture Galleries | Food and Drink Video
HOME » FOOD AND DRINK » RESTAURANTS

Bunga Bunga, London, restaurant review

Bunga Bunga in Battersea, London is a packed pizzeria that will steal your heart , says Keith Miller.

★★★★☆

Bunga Bunga, Battersea, London Photo: CLARA MOLDEN

By Keith Miller
7:00AM BST 13 Sep 2011

12 Comments

Bunga Bunga, 37 Battersea Bridge Road, London SW11 3BA,
Contact 020 7095 0360; www.bungabunga-london.com.
Price Dinner with wine, about £25 per head

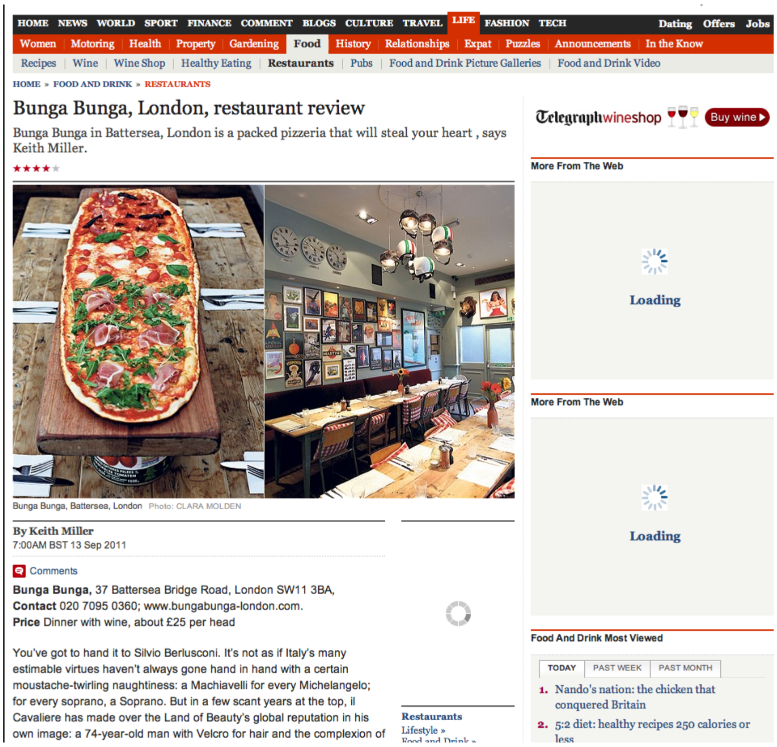
You've got to hand it to Silvio Berlusconi. It's not as if Italy's many estimable virtues haven't always gone hand in hand with a certain moustache-twirling naughtiness: a Machiavelli for every Michelangelo; for every soprano, a Soprano. But in a few scant years at the top, il Cavaliere has made over the Land of Beauty's global reputation in his own image: a 74-year-old man with Velcro for hair and the complexion of

Figure 5.3: The Telegraph Web page before the metadata was added

Looking at the figures we can see that the overall layout of the images is unchanged. There are however some changes to the HTML.

To make the Web page appear the same when viewed on the URL the user is given when the page is exported the relative URLs tied to images and CSS have been replace with absolute URLs. To keep the code simple only the URLs that start at the root of the domain were changed. So if there was an image on the page `http://example.org` which pointed to the source `/image.png` then the source would be changed to `http://example.org/image.png`. This change is not communicated to the user, which is unfortunate, but it is possible to change if it leads to issues.

As one can see from the images there are some sections of the Web site which are not displayed, or more precisely they are displayed as loading. The switch from relative to absolute URLs have only been done for images and CSS, which means that some of the JavaScript modules that the Web site depends on will not be loaded into the Web page. The tool tries to leave the HTML as much intact as possible, and since the scripts have yet seemed crucial to display the page properly it was decided to leave them as they were.


HOME NEWS WORLD SPORT FINANCE COMMENT BLOGS CULTURE TRAVEL LIFE FASHION TECH Dating Offers Jobs
Women | Motoring | Health | Property | Gardening | Food | History | Relationships | Expat | Puzzles | Announcements | In the Know
Recipes | Wine | Wine Shop | Healthy Eating | Restaurants | Pubs | Food and Drink Picture Galleries | Food and Drink Video
HOME » FOOD AND DRINK » RESTAURANTS
Telegraphwineshop
Buy wine
More From The Web
Loading
More From The Web
Loading
Food And Drink Most Viewed
TODAY
PAST WEEK
PAST MONTH
Restaurants
Lifestyle »
Food and Drink »

Bunga Bunga, London, restaurant review

Bunga Bunga in Battersea, London is a packed pizzeria that will steal your heart , says Keith Miller.

★★★★☆

Bunga Bunga, Battersea, London Photo: CLARA MOLDEN

By Keith Miller
7:00AM BST 13 Sep 2011

Comments

Bunga Bunga, 37 Battersea Bridge Road, London SW11 3BA,
Contact 020 7095 0360; www.bungabunga-london.com.
Price Dinner with wine, about £25 per head

You've got to hand it to Silvio Berlusconi. It's not as if Italy's many estimable virtues haven't always gone hand in hand with a certain moustache-twirling naughtiness: a Machiavelli for every Michelangelo; for every soprano, a Soprano. But in a few scant years at the top, il Cavaliere has made over the Land of Beauty's global reputation in his own image: a 74-year-old man with Velcro for hair and the complexion of

1. Nando's nation: the chicken that conquered Britain
2. 5:2 diet: healthy recipes 250 calories or less

Figure 5.4: The Telegraph Web page with metadata added

It should be noted that this means that the page will sometimes miss some content, as visible in the figures.

One issue that might arise is if some of the content on the site, either CSS or JavaScript, depend on the order of the elements or their direct placement in the hierarchy. Since MaDaME is intended to add metadata to user selected sections of the page the system had to to create its own HTML elements to attach the metadata to, since that is the only reliable way to talk about just that section. There are CSS selectors that select either direct children or the n-th children of an element. Since the system needs to add elements it might be that the element that was added cause another element to either not be the direct child of the former parent, or that it changes the elements number amongst the children. In a similar way, DOM manipulation in JavaScript sometimes uses the children of an element or the index of an element in a list of children. I have not found any way to avoid this issue, and can only hope that most developers depend on id and class attributes instead of the specific location in the DOM.

5.4 Analysis of usage data

In addition to the testing described earlier in this chapter the Web application has also been released on the Internet. This has allowed us to see how users who do not know the system would interact with it, and if they could use it to add metadata.

The data that has been gathered at this point is inconclusive, but can hopefully give some indication of how the tool will be used. The usage data that has been collected contains the IP of the user. This information was used to exclude users from the University of Bergen IP-range, as this data would mostly consist of usage tied to testing or debugging the system in the development phase. The usage data consists of the data from the 8th of April to the 12th of May.

As mentioned, the tool is mainly targeted towards adding metadata to single words, with adding data to sections being a secondary means of input. This is consistent with how the users have used MaDaME. From the usage data collected we can see that of the 298 of the text selections requests the server has received, only 12, or about 4%, have been for selections so big that they have been categorized as sections of text. The other 96% of the selections were of single words or of single entities or concepts like "Alexander Graham Bell" or "semantically classified lexical databases".

I have not found any cases of the users have used the resulting HTML on their Web pages, so it is not possible to say anything about their satisfaction with the page at this point.

Chapter 6

Summary and Conclusion

Before starting the work with this thesis I saw that there was a need for a system that would lower the barrier of entry to added semantic metadata to Web pages. I believe that having a Web of linked semantic data will become increasing important to find relevant information as the amount of information on the Internet continues to grow.

The advent of schema.org has made it easier to embed metadata on Web pages. The scope of the ontology however is largely limited to commercial and search engine specific concepts. In addition to this the microdata format pushed by the authors of schema.org does not allow for mixing vocabularies. I wanted to improve the situation by making it possible to add metadata about arbitrary content. My goal when starting work with this thesis was to create a prototype of a system that would allow users to add metadata to Web pages by using natural language, without requiring the to know the formal underpinnings of ontologies.

A research question was formalized that stated:

"Is it possible to create a tool which allows naïve users to easily add metadata to their Web sites using natural language?"

To answer this question there a number of sub-questions that has to be answered. How should users pick the parts of a Web page they want to add metadata to, and find the concepts it describes using natural language? Is WordNet suitable for representing disambiguated concepts from natural language in a way that will allow us to map these concepts to formal ontologies? How should an algorithm be implement to finds mappings from the natural language concept to types in formal ontologies in a way that preserves the semantic content of the concept? Is it possible to add metadata to Web pages in such a way that it does not change the way the page is rendered by browsers?

In this thesis I attempted to use WordNet as a method of representing natural language. WordNet was developed to have word boundaries corresponding to how humans mentally represent concepts. Using WordNet also made it possible to build on earlier work on creating mappings between WordNet and formal ontologies for the Semantic Web. WordNet also contains the concept of the hypernym, a relation that could be utilized to find mappings to higher level concepts if the synset itself did not contain a direct mapping to an ontology.

I examined the two algorithms for finding the best mapping from a given synset into the ontologies the system was mapping to. These were later evaluated to find which of them gave the best mappings.

A Web application was created to lets users add metadata to Web pages by selecting content on the page, and disambiguating the selection by clicking on suggested interpretations of the selection. MaDaME also allows users to import and export Web pages into the Web application for mark up.

6.1 Findings

I will now summarize the results of the analysis that was done in chapter 5.

The results found to a large degree supports the feasibility of using WordNet as a way to represent natural language when mapping to formal ontologies. Some cases where the integrity of the hypernym relation was violated were found. These errors have been reported to the maintainers of WordNet, and they will be fixed in the next public release (C. Fellbaum, personal communication, May 1, 2013 and May 13, 2013). I found that WordNet was unable to capture the grammatical number of natural language. This means that the tool will not have any way to distinguish between ontological types that differ in this respect. There were only a few instances in the ontologies that were utilized in the thesis of this flaw hindering the completeness of the mapping.

It was found that using hypernyms as a basis for mapping gave correct mappings. The incorrect mappings that were discovered during analysis were found to be caused by errors outside of the system. One should continue to look for further discrepancies in the future to help the development of these tools as well. The hypernyms first approach does however result in high-level mappings, and could benefit from further refinement.

The metadata that was created using MaDaME is comparable to that present at current Web sites which use schema.org to enrich their content. The testing also showed that the tool could help users avoid using incorrect types and properties, since it limits the properties allowed to add to those that are defined as belonging to the type.

As described in section 5.3 the testing of how the Web pages were rendered after metadata was added by the tool showed that the documents were displayed in the same manner before and after metadata was added. My analysis did uncover cases in which adding metadata could potentially change how the page was displayed. The tests did however find that the system was able to add metadata to the Web pages without changing the way they were displayed in the browser.

6.2 Further work

The goal of this thesis has been to create a functioning prototype of an artefact that allows users to add metadata to Web pages by using natural language. The system has been able to fulfill the most basic requirements, and shown that the concept is feasible. There are now several new interesting ways the tool could be developed further to increase its value to users, and to researchers.

It would be interesting have mappings to more ontologies, and one could offer the user a chance to say what the topic of the page was. In this way one could offer mappings to the Friend Of A Friend ontology if it was a Web page dealing with social interaction, the Good Relations ontology if it was a commerce page and so on. To do this the mapping module should get further development to complete the process of uncoupling the ontologies from the code.

There should be developed a way to allow for multiples of a single property on the Web page. The idea of adding properties came quite late in the project, and is as a consequence not as feature rich as it should be. One should also research into finding some way of allowing for properties from other ontologies. One difficulty here would be finding a way of presenting these without overwhelming the user.

Adding multiples might also mitigate the issue that synsets are not regarded as distinct because of their grammatical number. For schema.org the issue of aggregated terms is limited as it only has two types that are the aggregation of multiples of a type. By allowing multiples of properties, the system could handle adding the aggregation of these behind to the document automatically. This would be a good solution for the issue with schema.org, but it might not scale well if the system is expanded to include other ontologies that separate concepts by way of grammatical number.

When the Web site has experienced more usage it would be exciting to examine the usage logs to see which types of text gets tagged. Actual usage data would be an interesting source to discover concepts that users frequently want to map. Examination of these logs could therefore be a useful starting point to find out which ontologies to create mappings for, and which parts of these ontologies which would create the most value for the users. The usage data collected will make it possible to extrapolate text that the system did not find good disambiguations for by assuming that this is text that was selected but where the user did not click any of the suggested senses. It is also possible to count the frequencies at which different synsets or DBPedia terms were chosen as the concept the user wanted to describe, and use this to target the work of mapping.

6.3 Conclusion

My goal in this thesis was to answer the question if one could create a tool that let users add metadata to a Web page using natural language. I have now described the process of developing the prototype of MaDaME and the testing and analysis of the results. The findings have produced positive results for the main research questions. I have found a representational language that can capture the central semantics of natural language, and a means of mapping this language to ontologies with the help of mapping files. I have also managed to add the metadata to the Web pages without changing the rendering of the Web pages.

I acknowledge that there is still a lot of work that needs to be done before the system is finished. The system has traded expressiveness for simplicity, both to make it easier to use and to make the scope of the project manageable. The prototype has however been capable of answering my research question, and I feel confident that it has demonstrated the feasibility of creating a system that creates semantic metadata by utilizing natural language.

Bibliography

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). DBpedia: A Nucleus for a Web of Open Data. In Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., and Cudré-Mauroux, P., editors, *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg.

Auer, S. and Lehmann, J. (2007). What have Innsbruck and Leipzig in common? Extracting Semantics from Wiki Content. In *In ESWC*, pages 503—-517.

Bang, B. H. K., Dané, E., and Grandbastien, M. (2008). Merging semantic and participative approaches for organising teachers' documents. *Proc Conf. Educational Multimedia, Hypermedia & Telecommunications, pages*, pages 4959–4966.

Benzmüller, C. and Pease, A. (2012). Higher-order aspects and context in SUMO. *Web Semantics: Science, Services and Agents on the World Wide Web*, 12-13:104–117.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web - A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 284(5):34–+.

Brooks, C. H. and Montanez, N. (2006). Improved annotation of the blogopshere via autotagging and hierarchical clustering. *Proceedings of the 15th international conference on World Wide Web (S. 625-632). Edinburgh, Scotland: ACM.*

Chen, M., Fuhrt, B., and Purdin, T. D. F. (1990). Systems Development in Information Systems Research. *Journal of Management Information Systems*, 7:89–106.

Collins, A. M. and Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 8(2):240–247.

Fillenbaum, S. and Jones, L. V. (1965). Grammatical contingencies in word association. *Journal of Verbal Learning and Verbal Behavior*, 4(3):248–255.

Gantz, B. J. and Reinsel, D. (2011). Extracting Value from Chaos State of the Universe : An Executive Summary. Technical Report June, IDC.

Golder, S. and Huberman, B. A. (2005). The Structure of Collaborative Tagging Systems. *Audio, Transactions of the IRE Professional Group on.*

Gruber, T. (2007). Ontology of Folksonomy: A Mash-up of Apples and Oranges. *International Journal on Semantic Web & Information Systems*, 3(2):1 – 11.

Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing. In *International Journal of Human-Computer Studies*, volume 43, pages 907–928.

Guarino, N. (1998). Formal Ontology and Information Systems. In *FOIS '98: Proceedings of the international conference on Formal Ontology in Information Systems.* IOS Press.

Guha, R. (2011). Introducing schema.org: Search engines come together for a richer web. http://googleblog.blogspot.no/2011/06/introducing-schemaorg-search-engines.html.

Hebeler, J., Fisher, M., Blace, R., Perez-Lopez, A., and Dean, M. (2009). Modeling Information. In *Semantic Web Programming*, pages 63–92. Wiley, 1 edition.

Hevner, A. R., March, S. T., and Park Jinsoo, R. S. (2004). Design Science in Information Systems Research. *MIS quarterly*, 28(1):75–105.

Kim, H. L., Decker, S., Scerri, S., Breslin, J. G., and Kim, H. G. (2008). The state of the art in tag ontologies: a semantic model for tagging and folksonomies. *Proceedings of the 2008 International Conference on Dublin Core and Metadata Applications*, pages 128–137.

Mayer, M. and Menzel, J. (2009). More Search Options and other updates from our Searchology event. http://googleblog.blogspot.no/2009/05/more-search-options-and-other-updates.html.

Mika, P. (2005). Ontologies are us: A unified model of social networks and semantics. *The Semantic Web–ISWC 2005*, pages 522–536.

Miller, G. A. (1990). Nouns in WordNet: A Lexical Inheritance System. *International Journal of Lexicography*, 3(4):245–264.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Niles, I. and Pease, A. (2001). Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems - FOIS '01*, volume 2001, pages 2–9, New York, New York, USA. ACM Press.

Nunamaker Jr, J. F. and Chen, M. (1990). Systems development in information systems research. *System Sciences, 1990., Proceedings of the Twenty-Third Annual Hawaii International Conference on*, 3:631–640 vol. 3.

Passant, A. and Laublet, P. (2008). Meaning Of A Tag: A collaborative approach to bridge the gap between tagging and Linked Data. *Proceedings of the WWW 2008 Workshop Linked Data on the Web (LDOW2008), Beijing, China.*

Pemberton, S., Adida, B., McCarron, S., and Birbeck, M. (2008). {RDFa} in {XHTML}: Syntax and Processing. {W3C} recommendation, W3C.

Pretorius, A. J. (2004). Ontologies-Introduction and Overview. *Semantic technology and applications research laboratory.*

Ronallo, J. (2012). HTML5 Microdata and Schema.org. *The Code4Lib Journal*, (16).

Shirky, C. (2007). Shirky: Ontology is Overrated – Categories, Links, and Tags. http://www.shirky.com/writings/ontology_overrated.html.

Tang, J., Leung, H., Luo, Q., Chen, D., and Gong, J. (2009). Towards ontology learning from folksonomies. *Proceedings of the 21st international jont conference on Artifical intelligence*, pages 2089–2094.

Tennison, J. (2011). Lessons for Microdata from schema.org. http://www.jenitennison.com/blog/node/156.

Tonkin, E. and Guy, M. (2006). Folksonomies: Tidying up tags. *D-Lib*, 12(1).

Veres, C. (2011). LexiTags: An Interlingua for the Social Semantic Web. In *Proceedings of the 11th Interational Semantic Web Conference ISWC2011*, pages 1–12.

Weinberger, K. Q., Slaney, M., and Van Zwol, R. (2008). Resolving tag ambiguity. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia.*