

# A HYBRID APPROACH TO STATISTICAL AND SEMANTICAL ANALYSIS OF WEB DOCUMENTS

Thomas Gottron  
Johannes Gutenberg University  
Staudingerweg 9  
55128 Mainz/Germany  
gottron@uni-mainz.de

Roman Schneider  
Institute for German Language  
R5 6-13  
68161 Mannheim/Germany  
schneider@ids-mannheim.de

## ABSTRACT

This paper describes a new approach to improve the analysis and categorization of web documents using statistical methods for template based clustering as well as semantical analysis based on terminological ontologies. A domain-specific environment serves for prove of concept. In order to demonstrate the widespread practical benefit of our approach, we outline a combined mathematical and semantical framework for information retrieval on internet resources.

## KEY WORDS

Terminological ontology, template detection, IR

## 1. Background and Motivation

Even today, with the Web 2.0 as a reality and the Semantic Web being an evolving extension [1], information retrieval (IR) on the web is a challenging task. Semantic technologies offer new strategies for efficient web search, but often fail to deal with heterogeneous terminologies. Most WWW services tend to enrich documents with additional noise, due to navigation bars, blogrolls, or banner ads. In order to improve retrieval on multilingual or even interdisciplinary web content, we introduce a hybrid approach combining ontological knowledge with statistical noise filtering.

More than a decade ago, ontologies became a popular research topic in the fields of artificial intelligence, knowledge engineering, and IR. Though, the idea of describing relationships between real world objects or abstract topics was not new, it was often employed in different contexts and applications. Modern IR heavily relies on semantic add-ons for the classification and processing of distributed resources. The popular vision of a future "semantic web" will even force this trend. In order to establish language-independent frameworks, ambitious research activities within the knowledge engineering community deal with the modeling, coding, and linking of universal knowledge structures. Prominent examples of interdisciplinary developments are the *Suggested Upper*

*Merged Ontology (SUMO)*, *Cyc/OpenCyc*, the *Generalized Upper Model (GUM)* or *DOLCE/WonderWeb*<sup>1</sup>.

On top of these upper ontologies as well as stand-alone, more and more domain-specific ontologies are under construction. They codify concepts and relationships for single areas of interest, allow visualization and browsing of structures, and often include the goal of automated reasoning. For example, categories and relations dedicated to descriptive linguistics are captured with the help of *GOLD (General Ontology for Linguistic Description)*, which is built on top of *SUMO*<sup>2</sup>. Chiarcos [2] presents a way to integrate GOLD with different annotation models. However, even when limited to certain domains, ontology authors are faced with the simple fact that the terminological use of concepts vary between terminological systems. This seems especially true for linguistics, where different theories, schools, or authors often name concepts differently, or assign varying meanings to identical terms. E.g., generative grammars usually regard whole sentences as phrases, whereas others would categorize a phrase as a part of a sentence. Varying theories, varying timelines, varying analyzing criteria – creating a suitable backbone hierarchy would definitely provide considerable scientific benefit.

The heterogeneous use of terminology confuses human readers, and in the case of digitization makes information exchange between software systems on in human-computer interaction more difficult. Ontologies – often seen as enabling technology for information sharing – should cope with these difficulties. A semantically enriched IR application for the exploration of large domain-specific resources should "know" about theory-related details so that it can offer appropriate solutions. Schneider [3] introduces a way to deal with terminological differences and similarities. In order to bring together different systems, similar concepts are subsumed under so-called "termsets", so that the well-known synset paradigm used e.g. by *WordNet*<sup>3</sup> is expanded.

---

<sup>1</sup>Publication lists and technical information about these projects can be found at <http://www.ontologyportal.org>, <http://www.opencyc.org>, <http://www.purl.org/net/gum2>, and <http://wonderweb.semanticweb.org>.

<sup>2</sup> See <http://www.linguistics-ontology.org>.

<sup>3</sup> See <http://wordnet.princeton.edu>.

Another general problem for web based IR is the noisiness of web documents. Except the main article, they contain a lot of additional contents, such as navigation menus, copyright notices, commercials, etc. These additional contents are often template generated and contribute a lot to the content on the WWW. Gibson, Punera and Tomkins [4] estimated a ratio of approximately 40 to 50% of template generated content in modern web documents.

The approaches to clean web documents from this noise can be divided on the highest level into Content Extraction (CE) and Template Detection (TD) algorithms. CE algorithms use heuristics to locate the main content in a document. Typically they search text rich fragments in a web document. Solutions like the Document Slope Curves algorithm (DSC) [5] or Content Code Blurring (CCB) [6] fall into this category. TD, instead, chooses a different approach. TD algorithms are based on a set of documents which are all based on a common template. Learning the template structure from this training set, they construct a general pattern to locate the main content in all documents which are based on this template. The works of Bar-Yossef and Rajagopalan [7], the Site Style Tree algorithm [8] or the approach of Ma et al. [9] are representatives of this category.

Given the wide spread use of templates in web content management systems (WCMS), the TD approach is more suitable on web document corpora. The problem we have to deal with is to create suitable training sets for TD. In [10] we described a way to create such sets for a single document. In this paper we extend and modify this approach to handle a corpus of several documents.

Our primary motivation for combining the semantical with statistical template based approach was to improve IR for the *GRAMMIS* web information system<sup>4</sup>. *GRAMMIS* provides the most comprehensive specialist information about German grammar on the web.

The paper is structured as follows: we will first discuss the semantical approach, covering the steps of concept detection and relation modeling. The second part of the paper is dedicated to the statistical approaches to determine template groups in documents and the possibilities to clean a corpus of web documents from template generated noise. At the end we conclude the paper with a summary of our results and an outlook at future work.

## 2. Concept Detection

The first part of our hybrid approach comprises the building of a suitable terminological ontology. Apart from the modeling of relationship types, which is described subsequently, the selection of concepts is probably one of the most challenging subtasks within the ontology lifecycle. Concepts are chosen because of their relevancy to express

the knowledge in a given domain. Basically, they can be discovered by three different methods<sup>5</sup>:

1. Intellectual/manual compilation of all relevant domain concepts by human experts.
2. Use of statistical methods and automatic/semi-automatic algorithms on a given representative domain-specific corpus.
3. Use of linguistic methods.

Usually, the selection depends primarily on project-specific factors, preferences, and objectives. Recourse to human knowledge demands a relatively large amount of time, but generally guarantees high quality. Statistical methods depend on sufficiently large corpora as well as on long-time experience in fine tuning algorithms and parameters. Linguistic methods, e.g. the use of morpho-syntactic information, succeed only if parser, tagger, and lexicon supply reliable results.

The concept detection for the *GRAMMIS* ontology is based on a successful combination of statistical exploration, linguistic analysis as well as manual post-editing. The underlying specialist language corpus was made up of XML-structured hypertexts from grammatical web information systems and online dictionaries<sup>6</sup> hosted at the Institute for German Language. Altogether we included a total of about 2.000 hypertext nodes with 1.000.000 wordforms ( $N_{SL}$ ) and 45.000 tokens respectively. Furthermore we used *COSMAS (Corpus Search, Management and Analysis System)*<sup>7</sup> for exploring 160 large-scale general language corpora with more than 1.6 billion wordforms ( $N_{GL}$ ). As a result, we identified 3.500 wordforms – distributed over 1.400 concepts and 1.200 termsets – that seem relevant for our grammatical terminology. The concepts were acquired in the following six steps:

1. Frequency analysis of specialist language corpus: The specialist language (SL) hypertexts are used as input. We tokenize the corpus and collect frequency information for each token ( $f_{SL}$ ). Stop words are omitted. Wordforms with a frequency value below a previously defined threshold are filtered out. Output is an ordered list with two columns (wordform,  $f_{SL}$ ).
2. Markup analysis: This list from step 1 and XML-coded meta information from the grammar corpus serve as input. Wordforms appearing in the most prominent hypertext structures (i.e. in titles, subtitles, definitions, and semantically typed hyperlinks) receive a ranking bonus. Output is an accordingly modified  $f_{SL}$  list.
3. Frequency analysis of general language corpus: We use the output list from step 2 together with the *COSMAS*-maintained general language (GL) corpora as input. For each wordform, we calculate the GL-frequency value ( $f_{GL}$ ). Output is a list with three columns (wordform, modified  $f_{SL}$ ,  $f_{GL}$ ).

---

<sup>5</sup> Staab and Studer [11] offer comprehensive overviews of ontology engineering strategies.

<sup>6</sup> See e.g. Schneider [12].

<sup>7</sup> Online available at <http://www.ids-mannheim.de/cosmas2>.

---

<sup>4</sup> See <http://www.ids-mannheim.de/grammis>.



grouping of co-hyponyms, is linked with its hyperonym termset by a BTG relation.

In order to clarify the benefit of linking not only termsets, but also individual concepts, our example illustrates the relationships between *Phrase* and *Sentence*. Basically, the corresponding termsets are connected with the help of a BTP relation. Further, since generative grammars usually classify sentences as phrases, only these two concepts – singled out by a theory-related attribute – are linked by a *Narrower Term Generic* (NTG) relation (hyponymy). This fact, explicitly coded within the ontology base, should facilitate communication between people – or applications for the analysis of web documents/weblogs – using different terminological vocabularies.

Furthermore, we use standard relationship types like *Related Term* (RT) for the linking of termsets that are associated in some way, but without the necessity of deeper relationship explanation. Good examples are *Vocabulary* and *Vocabulary extension* or *Focus* and *Focusing adjunct*: Focusing adjuncts like "only", "even", "also" mark the focus. Because we do not see a need for introducing a special type for this relation, we simply call them RTs.

#### 4. Template Clustering and Detection

We mentioned already in the motivation the problem of noise in web documents. The high document frequency of words in additional contents, such as navigation menus, copyright notices, related link list, etc., might confuse algorithms and IR models based on term frequency. One reason for the constantly growing amount of noise in web documents is the increased use of templates [4]. Templates can be seen as document frameworks which are filled with different contents by a WCMS.

Hence, when operating on a corpus of web documents, it is necessary to apply methods to determine the main content. In this way we can eliminate and ignore the noisy parts of the documents. For template based documents, TD is the most suitable approach to this task. TD algorithms use a set of documents to derive a common template framework. Knowing the template induced parts of a document, they conclude the rest of the contents to be main content. These methods usually perform very well, but are in need of a clean and high quality training set.

##### 4.1 Detecting Different Templates in the Corpus

In a corpus of web documents we very likely have different templates. For each of these templates we need to construct a separate training set. Hence, we need to form groups of documents in our corpus which share the same template. In [14] we showed that a hierarchical clustering using single linkage based on a distance measure involving the longest common tag subsequence is most suitable to form such groups.

So, given an initial set  $C$  of documents we divide this set into  $n$  pairwise disjoint clusters  $C_i$ . Each cluster contains only documents which are all based on the same template. As soon as the clusters are computed, we can use them as

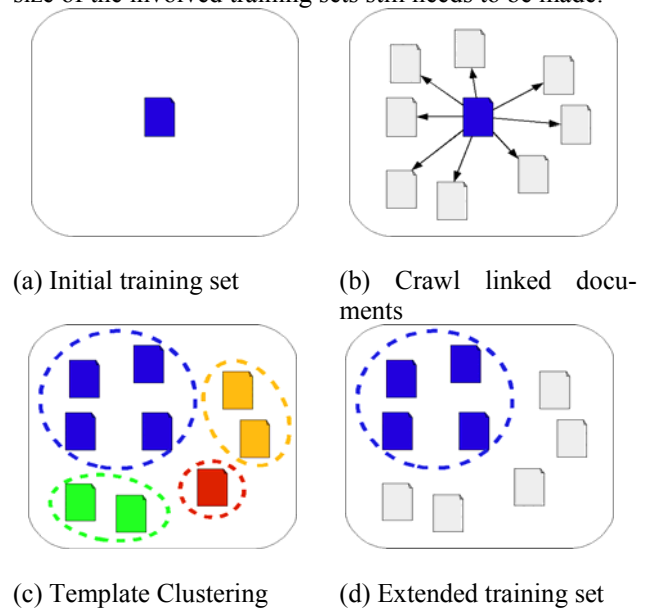
training sets for TD and deduce the underlying template framework. So, for each cluster  $C_i$  we learn an extraction pattern  $E_i$  and apply it to all documents in the cluster.

Moreover, in a constantly growing corpus, where new documents are added continuously, we can reuse the patterns. We just need to check if a new document falls into any of the known clusters. If this is the case, we know the document to be based on a known template for which we already have an extraction pattern at hand.

The text fragment frequency (TFF) based TD algorithm of Ma et al. [9] and the Site Style Trees (SST) [8] are both well documented and suitable for the template based web documents. For sake of brevity we omit the details of the algorithms and refer to the original papers.

##### 4.2 Extending Small Training Sets

In order to obtain good results from TD, the training sets need to be sufficiently large. Depending on the algorithm, TD authors provide different indications on how large a training set should be. To our knowledge, an extensive study on how TD algorithms perform depending on the size of the involved training sets still needs to be made.



**Figure 3: Extending small training sets with a local crawler**

Empirically we found the TFF based algorithm to provide satisfactory results with training sets of at least 30 to 40 documents, while SST provides quite good results already after being trained on 20 documents. So, an open question is how to deal with template clusters of less than 20 documents. In these cases we do not have enough documents to form a suitable training set<sup>10</sup>. Our solution follows the approach developed in [10] and extends small training sets (not the corpus) with suitable documents from the web.

<sup>10</sup> A too small training set also occurs if a new document with a previously unseen template is added to the corpus.

The process is outlined in figure 3. Starting from the documents in the training set (a), we collect all documents they reference by hyperlinks (b). Documents created by a WCMS usually contain a high rate of intra-site links. Hence, by following the hyperlinks we tend to find a lot of documents from the same website, which in turn are very likely to be based on the same template. To separate documents based on other templates we use again a structure based cluster analysis (c). Finally, we extended the small training set by those documents which lie in the same cluster as the initial training documents (d).

### 4.3 Guiding the Crawler

To speed up this extension process it is possible to guide the crawler which collects the linked documents. We exploit that documents with a similar URL tend to be based on the same template. An explanation for this observation is that the URLs of a web site often reflect some hierarchical structure which corresponds to the used templates. Thus, it is possible to forecast a certain structural similarity for two documents if their URLs are similar. Therefore, we guide the crawler to follow those URLs first which are similar to the URL of the initial document. We considered several URL similarity measures for this task. Wang and Zaïane [15] developed an URL similarity measure (**Wang Zaïane**), for which they split the paths of URLs into tokens and calculate a weighted sum of common path elements, giving higher importance to tokens that appear earlier in the path. We extended this measure to host and parameter data (**Extended Wang Zaïane**), because otherwise two URLs might have a perfect similarity of 1 even if they are originating from entirely different web sites (e.g. entry pages like *index.html*) or if they have different parameters which might affect the appearance (e.g. the presence of a parameter to indicate an entirely different but printer friendly layout).

While the above weighing scheme seems intuitively a good approach, it has not been analyzed further. Ignoring the weights corresponds to measuring the similarity based on the **Common Path** of two URLs.

Considering only the common prefix of the URLs is another unnecessary restriction. If URLs share a common suffix or large infix after the first differing token they are still similar and might hint to similar templates. Taking into account the longest common subsequence of tokens (**Token LCSS**) follows this way of thought. The **LCSS** similarity even skips the tokenization and computes the longest common subsequence on the full URL strings.

To find out which URL similarity measure in general reflects best the expected template similarity of the documents we conducted a simple experiment. We randomly selected 773 documents from web sites listed in the DMOZ. DMOZ entries are often the entry pages of a web site. Hence, they frequently lack a path in their URL. Further, entry pages typically have their own particular template. For this reason, we followed a random link on the entry page to penetrate deeper into the web site hierarchy and chose documents with more than 20 hyperlinks for

our experiment. Then we analyzed which of the hyperlinks lead to documents with a high structural similarity and considered those URLs relevant for our training set.

We implemented several crawlers, which used the different URL similarity measures to determine which hyperlinks to follow first. This means, we sort the URLs of the hyperlinks according to their similarity to the URL of the initial documents. We then crawl them from the most to the least similar. As a baseline for the guided crawlers we used a crawler following the URLs in a random order. The recall precision graph for this experiment is shown in figure 4. The curves show the precision of the result sets when achieving certain recall values.

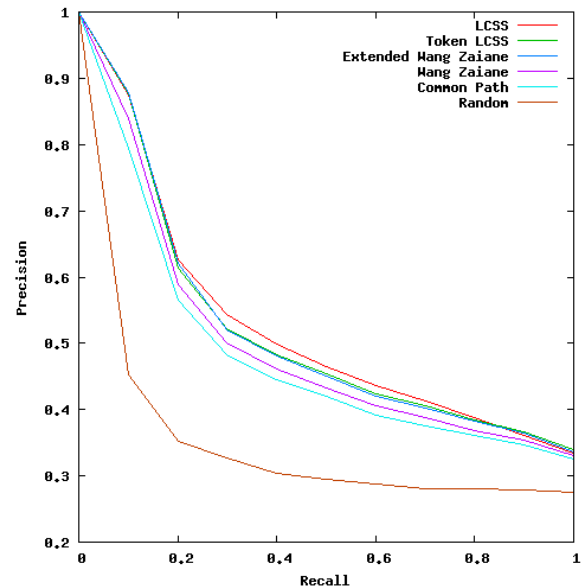


Figure 4: Recall precision graph for the guided crawlers.

All curves are significantly higher than the random crawling process, i.e. each guidance tends to crawl documents first which are more likely to be based on the same template. The original Wang Zaïane measure is doing better than the unweighted Common Path variation, confirming the intuitive introduction of higher weights for tokens which appear earlier in the path. The Extended Wang Zaïane approach is comparable to the measuring the longest common subsequence of the tokens, and is outperformed only by using the longest common subsequence of the URL string representation to calculate a similarity.

It seems surprising at first sight, that the character based LCSS outperforms the token based methods. An explanation might be that the token based approaches suffer from small changes in the URL. E.g. if an URL contains the path */us\_politics/* and another one */eu\_politics/*, the tokens for this part of the URL are different while the strings are quite similar. This string similarity might also indicate similar topics and similar templates, after all.

### 4.4 Fallback Solution: Content Extraction

In most cases this process provides enough documents to build training sets of sufficient size – even if starting from

a single document. If this is not the case, the extension of the training set can be repeated iteratively several times.

If the number of documents based on a particular template still is too small to apply TD (e.g. because we have to deal with a document that is simply not based on a template), we have three options.

The first is to exclude the documents from the corpus. This is a very aggressive option as it prefers to omit data rather than deduce wrong conclusions for relevance information or ontology construction. Alternatively we can use the documents in their original form. Though we said, that template generated contents might cause distortions in term frequency values, this might be negligible in this particular case. As we found few documents for a certain template, its template generated contents will not bias the term frequencies very strong (which is valid in particular for large corpora). Finally, we can apply single document CE heuristics which can reduce noise without the need of a training set. DSC [5] and CCB [6] are very good algorithms for identifying text based main content.

The last option seems to be the best compromise, because we do not exclude documents from the corpus (which is of importance for IR tasks) while keeping the risk of noise as small as possible.

## 5. Conclusion and Future Works

Our hybrid approach already allows the integration of different terminological systems and languages with language-independent statistical methods, and thereby supports international collaboration and research. We believe that multilingual, theory-spanning domain ontologies will be a clear asset for the analysis of web documents and – more generally – for all projects related to the vision of the semantic web. Our aim is not so much the formal unification of ontological models, but rather the accurate representation of domain-specific concepts and relationships with respect to varying retrieval and classification goals. We accept that there is no self-evident way of dividing the world – or even small parts of it – into concepts. Especially in terminology we often deal with hardly dissolvable antagonisms.

Statistical methods for the clustering of template based web documents extend our terminological analysis by adding strategies for identifying the main content within documents. This is necessary to clean documents from noise. In general, with the methods described here we are well prepared for cleaning a corpus of web documents. We use field tested TD algorithms for our purpose and a reliable mechanism for training set creation. The extension of too small training sets is realized efficiently by using guided crawlers and we incorporate a fallback system of heuristics.

The guidance of the crawlers and the performance of the TD and CE algorithms leaves space for optimization and is part of further research.

## References

- [1] R. Schneider, Web 3.0 ante portas? *Kommunikation, Partizipation und Wirkungen im Social Web*, Köln: Herbert von Halem Verlag, pp. 112-128.
- [2] C. Chiarcos, An Ontology of Linguistic Annotations. *GLDV Journal for Computational Linguistics and Language Technology*, 23(1), 20008, 1-17.
- [3] R. Schneider, Frequency & Markup Analysis for Terminological Ontologies. *Proceedings of the Workshop on Exploiting Semantic Annotations for Information Retrieval*, 2008, 83-87.
- [4] D. Gibson, K. Punera, A. Tomkins, The Volume and Evolution of Web Page Templates. *Proceedings of the 14<sup>th</sup> International Conference on World Wide Web*, 2005, 830-839.
- [5] D. Pinto, M. Branstein, R. Coleman, B. Croft, M. King, W. Li, X. Wei, QuASM: a System for Question Answering Using Semi-Structured Data. *Proceedings of the 2<sup>nd</sup> ACM/IEEE-CS joint conference on Digital libraries*, 2002, 46-55.
- [6] T. Gottron, Content Code Blurring: A New Approach to Content Extraction. *Proceedings 5<sup>th</sup> International Workshop on Text Information Retrieval*, 2008, 29-33.
- [7] Z. Bar-Yossef, S. Rajagopalan, Template Detection via Data Mining and its Applications. *Proceedings 11<sup>th</sup> International Conference on World Wide Web*, 2002, 580-591.
- [8] L. Yi, B. Liu, X. Li, Eliminating Noisy Information in Web Pages for Data Mining. *Proceedings of the 9<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003, 296-305.
- [9] L. Ma, N. Goharian, A. Chowdhury, M. Chung, Extracting Unstructured Data from Template Generated Web Documents. *Proceedings 12<sup>th</sup> international Conference on Information and Knowledge Management*, 2003, 512-515.
- [10] T. Gottron, Bridging the Gap: From Multi Document Template Detection to Single Document Content Extraction. *Proceedings of the IASTED Conference on Internet and Multimedia Systems and Applications*, 2008, 66-71.
- [11] S. Staab, R. Studer, (eds), *Handbook on Ontologies*. (Springer Series on Handbooks in Information Systems. Berlin: Springer, 2004).
- [12] R. Schneider, E-VALBU: Advanced SQL/XML processing of dictionary data using an object-relational XML database. *International Journal for Language Data Processing*, 32(1), 2008, 35-46.
- [13] L. Gillam, M. Tariq, K. and Ahmad, Terminology and the construction of ontology. *Terminology Vol. 11(1)*, 2005, 55-81.
- [14] T. Gottron, Clustering Template Based Web Documents. *Proceedings 30<sup>th</sup> European Conference on Information Retrieval*, 2008, 40-51.
- [15] W. Wang, O.R. Zaïane, Clustering Web Sessions by Sequence Alignment. *Proceedings of the 13<sup>th</sup> International Workshop on Database and Expert Systems Applications*, 2002, 394-398.