Contextual Approaches for Identification of Toponyms in Ancient Documents

Hendrik Schöneberg and Frank Müller
Institute of Computer Science, University of Würzburg, Würzburg, Germany

Keywords: Named Entity Recognition, Information Retrieval, Contextual Approach, Feature Vector, Conditional Random Field, Classification Framework, Digitization.

Abstract: Performing *Named Entity Recognition* on ancient documents is a time-consuming, complex and error-prone manual task. It is a prerequisite though to being able to identify related documents and correlate between named entities in distinct sources, helping to precisely recreate historic events. In order to reduce the manual effort, automated classification approaches could be leveraged. Classifying terms in ancient documents in an automated manner poses a difficult task due to the sources' challenging syntax and poor conservation states. This paper introduces and evaluates two approaches that can cope with complex syntactial environments by using statistical information derived from a term's context and combining it with domain-specific heuristic knowledge to perform a classification. Furthermore, these approaches can easily be adapted to new domains.

1 INTRODUCTION

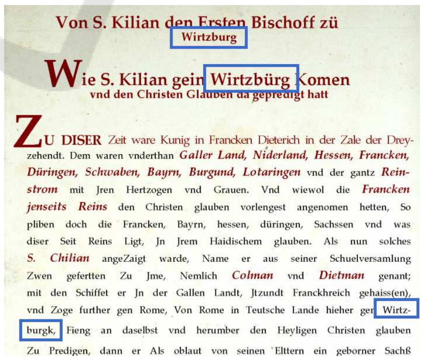
Digitization projects like the *Google Books*[1] initiative have massively grown in scale within the last years, seeking to offer documents to a wide audience. The digitization process usually obtains a mere digital copy of a given source - i.e. a set of images. In order to retrieve the source's semantics as well you have to put in extra effort. This might begin with retrieving a textual representation of your source (via *OCR* or even manually) and includes identification and persistence of important / interesting terms or passages.

This kind of information retrieval - often referred to as *Deep Tagging* - is a necessity in order to be able to link to other related documents, to correlate between terms in distinct documents and, generally said, to provide better results for users' queries.

Automated approaches that help with this task yield good results as long as the input document has a continuous structure, is well-conservated and its digitized copy is of high quality. Furthermore, for automated classification approaches the source's language and domain should be well-known. But what if one or more of these preconditions are not met?

Ancient Documents. For several reasons ancient documents pose a special challenge for digitization and deep tagging procedures.

[1]http://books.google.com


Von S. Kilian den Ersten Bischoff zü
Wirtzburg

Wie S. Kilian gein Wirtzbürg Komen
vnd den Christen Glauben da gepredigt hatt

ZU DISER Zeit ware Kunig in Francken Dieterich in der Zale der Dreyzehendt. Dem waren vnderthan *Galler Land, Niderland, Hessen, Francken, Düringen, Schwaben, Bayrn, Burgund, Lotaringen* vnd der gantz *Reinstrom* mit Jren Hertzogen vnd Grauen. Vnd wiewol die *Francken jenseits Reins* den Christen glauben vorlengest angenomen hetten, So pliben doch die Francken, Bayrn, hessen, düringen, Sachssen vnd was diser Seit Reins Ligt, Jn Jrem Haidischem glauben. Als nun solches *S. Chilian* angeZaigt warde, Name er aus seiner Schuelversamlung Zwen gefertten Zu Jme, Nemlich *Colman* vnd *Dietman* genant; mit den Schiffet er Jn der Gallen Landt, Jtzundt Franckhreich gehaiss(en), vnd Zoge further gen Rome, Von Rome in Teutsche Lande hieher gen Wirtzburgk, Fieng an daselbst vnd herumber den Heyligen Christen glauben Zu Predigen, dann er Als oblaut von seinen Elttern ein geborner Sachß

Figure 1: Fries Chronicle, 16th century, rendered transcript with exemplary orthographic issues highlighted.

Figure 1 shows a rendered transcript of the *Fries Chronicle*[2], a chronicle from the 16th century written in *Early New High German*. Notice how the figure shows three different spelling variants for the German city Würzburg, even though their occurrence is only a few lines apart. Capitalization of characters was mainly subject to the author's assessment of the term's importance in its current context. Depending on personal preference some authors even chose to capitalize single letters, just because they took a liking to it.
Poor conservation states add to the complexity of data mining within ancient sources, as any information

[2]http://franconica.uni-wuerzburg.de/ub/fries/index.html

Schöneberg H. and Müller F..
Contextual Approaches for Identification of Toponyms in Ancient Documents.
DOI: 10.5220/0004110701630168
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2012), pages 163-168
ISBN: 978-989-8565-29-7

contained within damaged parts can only be reconstructed heuristically. These factors lead to a high variety of spelling variants (see Table 1).

Table 1: Spelling variants for term 'Würzburg'.

Spelling variants for 'Würzburg' (excerpt)
Herbipolis, Wirceburgum, Wirciburc, Wirtzburg, Wirzburg, Wirziaburg, Würtzburg, Würtzb, wurtzb, Wurtzburg, wurtzburg, würtzburgk, würtzburg, W, würtzberg, Wurtzb, Würzburg, wurzburg . . .

Identifying all spelling variants for a given *Named Entity* in an automated manner is a non-trivial task, in which traditional dictionary-driven information retrieval and markup approaches can only succeed to a certain degree, as it is very unlikely that all its spelling variants have been discovered yet.

Goal. We need to find classifiers for *Named Entities* with the ability to cope with syntactical challenging environment and a broad variety of spelling variants.

2 CONTEXT CLASSIFICATION APPROACHES

As the previous section pointed out, the domain of ancient documents poses a special challenge for *Named Entity Recognition* due to the terms' high variability. But instead of focusing on compensating for the terms' high variability, couldn't we instead look out for more reliable sources of information?

Term Co-occurrence. Contrary to the loosely defined orthography, the grammar and sentence structure in ancient documents are comparably restrictive as nowadays. Thus the likelihood of two terms t_1, t_2 co-occurring is not arbitrary but instead has a specific probability.

Stop Words. A term's context mostly comprises *stop words*. Stop words are terms that mainly fulfill a linguistic purpose and do not carry much information themselves (as prepositions, conjunctions or articles). Due to the stop words' frequent occurrence their orthographic consistency is much higher than that of *Named Entities* like places or people's names.

'Event-driven' Tagging. As already pointed out, stop words carry little information apart from their linguistic function. In our task to extract a source's semantics they can obviously be neglected. But even within the group of non stop words only a small subset is relevant to our interests: Thinking of Wiki-Systems like Wikipedia [1], usually only a small subset of *Named Entities* is cross-referenced, for example other people's names, people's function, role or profession (like mayor), places or dates. In order to precisely recreate past events it is usually necessary to find out, *who* did something, *when* did he / she do it and *where*. We therefore define an *Event e* as $e = (a, d, p)$ with $a \in A$ and A being the set of all actors, $d \in D$ with D being the set of all dates and $p \in P$ with P being the set of all places. We can now reduce the complexity of *Named Entity Recognition* on ancient sources by limiting the relevant classes to events or integral parts of events.

2.1 Related Work

According to (Miller and Charles, 1991), the exchangeability of two terms within a given context correlates with their semantic similarity. This means, the easier two terms are exchangeable within the contexts they occur, the more likely they share a similar meaning. A statistical analysis of two terms' context composition can therefore indicate their degree of semantic similarity. Many approaches utilize the information contained within a term's context: (Gauch et al., 1999) propose an automatic query expansion approach based on information from term co-occurrence data. (Billhardt et al., 2002) analyze term co-occurrence data to estimate relationships and dependencies between terms. (Schütze, 1992) uses contextual information to create *Context Vectors* in a high-dimensional vector space to resolve polysemy.

Existing Knowledge. Of course, any *a-priori* knowledge aggregated in databases for domains like toponyms (names derived from a place or region), professions or male names and their spelling variants is not discarded, but used as the entry point for our contextual approach, as it reliably shows us instances of our sought-after class. We can then analyze their contextual properties to find additional, so far unknown instances. Furthermore we can inspect the a-priori data for useful patterns to create domain specific heuristics (e.g. typical n-gram distribution for a given class, typical pre-/suffixes, capitalization).

As a term's context can apparently provide information about its semantics, the following sections therefore introduce approaches that attempt to classify a term as an integral part of an event by evaluating its contextual information.

[1]http://en.wikipedia.org

3 CONTEXT VECTOR ANALYSIS

The concept behind the context classfication approach presented in this section is *learning by example*: Starting with instances of our sought-after class we try to derive their common contextual properties. You can then scan the document for terms with similar contextual properties and propose that those terms share the same class.

This procedure shall be demonstrated step by step.

3.1 Creating the Reference Profile

Initially you are required to specify instances of the class you want to classify, further referred to as *reference terms*. All occurrences of the reference terms will be located throughout the source.
Each occurrence of a reference term has a *context* in which it is used and it is the *contexter*'s responsibility to extract the term's context. Several contexter implementations have been evaluated, but *fixed window size* approaches turned out most useful. That means, that a term t's context comprises w terms / chars left and / or right of t.

3.2 Analyzing the Context

The crucial part of the classification process is the analysis of a context's properties. For this purpose it is passed to a pipeline of *feature modules*, in which each module will output a list of properties for the given context (see Figure 2).

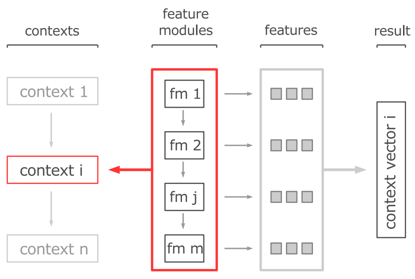


Figure 2: Analysis of context properties.

The feature modules can be of either *statistic* (see Table 2) or *heuristic* (see Table 3) nature.
The biggest advantage of analyzing a context with *statistic modules* is that this works independently from the source's domain, whereas *heuristic modules* add domain-specific knowledge to the analysis of contexts, which can be used to rule out *false positives* or affirm terms, which otherwise would have been discarded.

Table 2: Statistic feature modules (excerpt).

Context Composition	Extracts information on how the context is composed, i.e. what terms occur
Co-occurrence	Derives information about which terms co-occur within the context
Distance metric	Outputs information about the distance in which a context term item appears related to the query term

Table 3: Heuristic feature modules (excerpt).

Part Of Speech	A term's part-of-speech information can be used to extract linguistic patterns from a context or help to eliminate false positives.
Pre-/Suffix heuristic	Provides hints, whether the currently analyzed context contains pre-/suffixes common for the class we wish to find.
Dictionaries	Domain specific dictionaries can provide an additional source of information (e.g. a list of all German cities, Spanish male names or professions in English language).
Semantic markup	If the source already contains any semantic markup (tags indicating a place, name, ...), this knowledge can be evaluated.

3.3 Aggregation and Comparison

All feature modules generate a list of features for a given context (see Figure 2).
Each feature can then be regarded as a dimension in feature vector space. After the pipeline of feature modules has analyzed a context, the resulting features will be summed up in a context profile, which is a vector in feature space and is finally normalized with the amount of contexts contained within. With this normalized context profile you now have a metric for contextual similarity. We can estimate the similarity of two context vectors by calculating their cosine angle. For further information on the *Vector Space Model* see (Baeza-Yates and Ribeiro-Neto, 1999).

3.4 Context Vector Classification Algorithm

Let D be a document, $T \subseteq D$ a set of reference terms sharing the same class, e.g. $T = \{London, Berlin, \ldots\}$. For each $t \in T$ do:

1. **Reference Step.** Let t have $count(t)$ occurrences within the source, for each occurrence t_i with $0 \leq i < count(t)$ do:

   (a) Retrieve *context* for term t_i (as described in Section 3.2)
   (b) Analyze and aggregate context properties into *context profile* C_T (as described in Section 3.3)

2. **Classification Step.** In order to classify a term $s \in D$, create context profile C_s accordingly. If the similarity of C_s and C_T falls below a threshold δ, propose that s has the same class as the reference terms from T.

   $$similarity(C_s, C_T) \leq \delta \Rightarrow class(s) = class(t_i), t_i \in T$$

   with $similarity(a,b)$ being the cosine angle as described in (Baeza-Yates and Ribeiro-Neto, 1999).

The second contextual classification approach leverages a similar methodology and will be presented in the following section.

4 CONDITIONAL RANDOM FIELD

The second classification approach uses *Conditional Random Fields* (CRF) to train a model able of predicting a term's class. CRFs are undirected graphical models and often applied in pattern recognition / labelling related tasks like natural language processing, structure prediction of DNA-sequences or object recognition. For further information see (Lafferty et al., 2001).

Preference over HMMs. A trained CRF represents a statistical model which is able to emit a *label* (read: classification) for a given observation, which strongly resembles the *Hidden Markov Model* (HMM). But in difference to the HMM the CRF emits a label not only depending on the model's current state and current observation, but instead takes into account the *adjacent* states as well and can at any time access the *entire* observation sequence. This means, that again we have the possibility to take into account a term's contextual information.

Learning by Example. The model is trained by comparing labels from a training set with the model's prediction and applying error-minimizing methods like the *Quasi-Newton method* or *Gradient descent*. After successful training we want the model's predicted sequence of labels for the given sequence of observations to match the sequence of labels from the training set in the best possible way. But how does the model know how to label a given observation?
The CRF is trained by example: We need to supply exemplary observation sequences and their respective labels. What makes CRFs valuable for our approach is its ability to process an unbounded number of features connected to each observation. A feature can be regarded as an observation's property and is crucial for the classification process. These properties have to be applied to the observation sequence in a preprocessing step and will be accounted for in the training procedure. Combined with the CRF's characteristic of being able to inspect the entire observation sequence at any time, this means, we can now - pretty much analogue to the context vector approach as described in Section 3.2 - analyze an observation's context with a pipeline of feature modules, each of which emits a list of features for the currently analyzed context.

Supervised Learning. In order to improve its overall performance, the approach relies on supervised learning strategies. Therefore the workflow contains a learning cycle loop incorporating user feedback to progressively optimize the training and efficiency of the underlying CRF. After the annotation of the source document, based on the applied set of features as described in Section 3.2 on page 3, a subset of the document, consisting of text passages that enclose classification examples for the sought-after class, is being composed. Upon entering the learning cycle, a CRF is being trained on the previously generated subset and immediately applied to the entire annotated source document. Whenever the CRF assigns a label of interest to a term, the user is provided with a proposal consisting of the term in question and its adjacent context. By accepting the proposal, the suggested label is applied to the focused term and the training set for the CRF is extended by either all of the terms occurrences within the document or just its current occurrence. This newly obtained, user-affirmed instance for a class can be added to the training data. Then, based on the updated training set, a new cycle begins with another training of the statistical model. The learning cycles continue until user termination or when a threshold of newly classified terms or their occurrences within the text is underrun.

5 RESULTS

The context classification approaches will be evaluated against two ancient documents. The first document is the *Fries Chronicle*[1] from the 16th century, with around 28.000 words in *Early New High German*. The second document is *Topographia Franco-*

[1]http://franconica.uni-wuerzburg.de/ub/fries/index.html

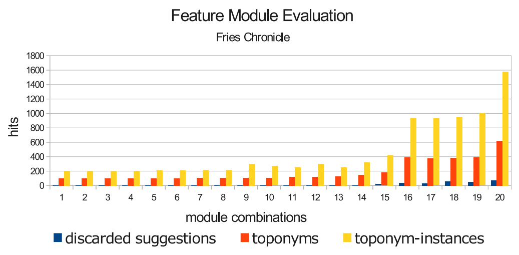


Figure 3: Toponym-instances vs feature module combination. Source: Fries Chronicle.

niae[2] from 1648, comprising 76.000 words. The classification framework was presented a set of known toponyms and challenged to identify as many new instances as possible.

Feature Module Combinations. As described in Section 3.2, the classification quality directly depends on the selection of feature modules analyzing the context. Figures 3 and 4 show exemplary test runs with arbitrarily selected feature module combinations. The x-axis denotes test runs with different feature module combinations, whereas the y-axis displays the amount of unique toponyms and absolute amount of toponyms that were found. As you can derive from the figures, some feature module combinations only yield mediocre results, while others perform much better. As an example: The good results seen on test run 20 in both figures resulted from combining the the statistic feature module analyzing term co-occurrence data and the heuristic feature modules responsible for part-of-speech information and semantic markup evaluation. The general directive is to figure out the best feature module combination for a given classification task (see Section 7).

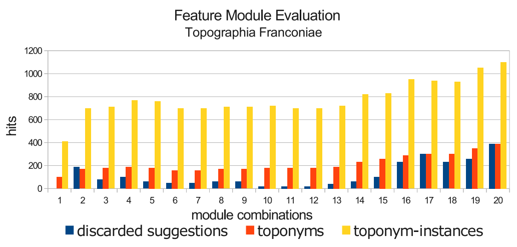


Figure 4: Toponym-instances vs feature module combination. Source: Topographia Franconiae.

Learning Cycle. Figure 5 illustrates the effects of the learning cycle as described in Section 4. For a given feature module selection, the CRF's suggestions are iteratively affirmed or declined by the user and the model is retrained with the newly learned data. Figure 5 shows a steep increase in the amount of toponyms found within the source, converging against its maximum after three iterations. More results can be found in (Müller, 2012).

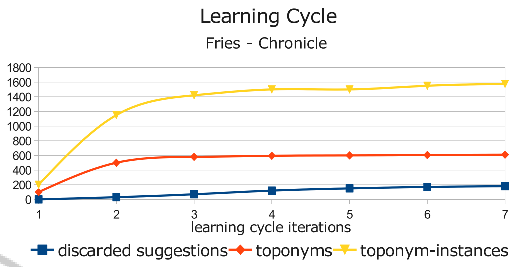


Figure 5: Evaluation of learning cycle.

6 EVALUATION

Our goal was to create a tool, supporting the user to classify terms within ancient documents in an automated manner. Ancient sources feature a very narrow, brittle domain (see Section 1), which greatly reduces the utility of dictionary- or rule-based approaches. On the other hand a single, precise language model for the term classification in any ancient document can not be applied, as the next document might feature a slightly different domain (for example due to regional distance, another author, etc).

A central requirement for a classification framework for ancient sources is therefore its easy adaptability to new domains. In section 2 we relaxed some of the initial constraints by reducing the number of classes we actually are interested in to just a few and specialized in finding *events* or integral parts of events like people's names and roles, places or time specifications. Our approach leverages statistical analysis of a term's context, which will work independently from the source's domain. The results can be optimized by using existing domain-specific, heuristical knowledge.

6.1 Advantages

Flexibility. Our contextual classifier is easily adaptable to changing domains as it on the one hand applies *statistic* feature modules, which work independently from the given source's language and domain and on the other hand uses *heuristic* feature modules, which can add domain specific knowledge to the classification procedure. Either set of modules can easily be extended / reduced to adapt to a given domain.

Polysemy. The presented approach is able to overcome some of the restrictions a traditional dictionary-

[2]http://franconica.uni-wuerzburg.de/ub/35a1128/index.html

or rule-based classification approach has to face. Depending on the context a term is used in, it can be decided whether it belongs to a certain class or not, thus effectively resolving polysemy.

Concept Classification. If you think of *London*, *Paris* and *New York*, these are obviously easily identifiable instances of the class *place*. The contextual classification approach even allows the identification of abstract, compound places like *'the well behind town'*. Another example: Aristocratic titles often state a person's *role* (like 'earl'), *name* (like 'George') and origin (like 'from London'), which effectively is a *place*. The compound title on its whole refers to a person and should therefore be classified as *name*. The contextual classifier can correctly identify these complex names.

Proposal System. Our contextual classification approach is not intended to be a replacement for dictionary- or rule-based approaches. Instead it should be combined with other classifiers within a framework to affirm / negate classification suggestions. Furthermore, based on contextual evaluation, it can suggest new, so far unknown instances. It is well suited to work within a *proposal system* or *expert system*.

7 FUTURE WORK

As the presented methodology is work in progress, it can be optimized and extended in many ways. The implementation provides a framework, focusing on modularity and extensibility, which guarantees that the involved parts can be exchanged easily. This section lists some possible extensions that could increase the overall classification performance.

Feature Module Optimization. The *feature modules'* (compare Section 3.2) performance is central to the classification quality. Besides increasing the number of feature modules it is therefore crucial to identify the modules, which yield the best results in classifying a certain class. As the output of feature modules analyzing a term's context for linguistic patterns in many cases is not *linearly separable* into distinct classes, it is hard for machine learning approaches to identify the subset of feature modules that work *best* for a given class. This problem could be solved by leveraging *genetic algorithms*.

Boosting. The current implementation of the classification framework utilizes both presented approaches and treats their outputs equally. Depending on the source or domain, either algorithm could yield better results than the other. According to (Schapire, 2002), *boosting* could provide the means to overcome this problem. By applying a boosting-algorithm to create a weighted compound output of the two approaches the overall classification quality for a given class could be improved.

REFERENCES

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. ACM Press, New York, 1st edition.

Billhardt, H., (corresponding), H. B., Borrajo, D., and Maojo, V. (2002). A context vector model for information retrieval. *Journal of the American Society for Information Science and Technology*, 53:236–249.

Gauch, S., Wang, J., and Rachakonda, S. M. (1999). A corpus analysis approach for automatic query expansion and its extension to multiple databases. *ACM Trans. Inf. Syst.*, 17(3):250–269.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, San Fransisco. Morgan Kaufmann.

Miller, G. A. and Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6.

Müller, F. (2012). Identifikation von Toponymen in historischen Texten. Master's thesis, Julius Maximilians Universität Würzburg.

Schapire, R. E. (2002). The Boosting Approach to Machine Learning An Overview. In *MSRI Workshop on Nonlinear Estimation and Classification*.

Schütze, H. (1992). Dimensions of meaning. In *Supercomputing '92: Proceedings of the 1992 ACM/IEEE conference on Supercomputing*, pages 787–796, Los Alamitos, CA, USA. IEEE Computer Society Press.