# CONTEXT VECTOR CLASSIFICATION
## *Term Classification with Context Evaluation*

Hendrik Schöneberg
*Institute of Computer Science, University of Würzburg, Würzburg, Germany*

Keywords: Text mining, Classification, Deep tagging, Information retrieval.

Abstract: Automated *Deep Tagging* heavily relies on a term's proper recognition. If its syntax is obfuscated by spelling mistakes, OCR errors or typing variants, regular string matching or pattern matching algorithms may not be able to succeed with the classification. *Context Vector Tagging* is an approach which analyzes term co-occurrence data and represents it in a vector space model, paying specific respect to the source's language. Utilizing the cosine angle between two context vectors as similarity measure, we propose, that terms with similar context vectors share a similar word class, thus allowing even unknown terms to be classified. This approach is especially suitable to tackle the above mentioned syntactical problems and can support classic string- or pattern-based classificator-algorithms in syntactically challenging environments.



## 1 INTRODUCTION

**Motivation.** Let us assume being researching an arbitrary topic via the internet. Unless we explicitly know a source that provides the sought-after information, at some point we'll most likely find ourselves having to use a search engine. The search engine's success depends heavily on the query we submit. Unfortunately, due to e.g. different educational backgrounds, language habits or personal preference people can express their ideas very differently. According to (Furnas et al., 1987) and (Deerwester et al., 1990) studies show, that only in less than 20% of the time two people choose the same keyword to describe a single, well-known object.

In an attempt to make an arbitrary source more accessible to a broad variety of search queries, it is of high interest to provide additional knowledge going beyond the source's intrinsic information. To name a few examples, this ranges from keywords describing the source's category of content, editorial information or cross-references to related articles, up to information with pin-point granularity like synonyms for a specific term. The process of annotating a source with this additional information is called *Deep Tagging*. Deep Tagging a source manually is a time-consuming and error-prone process if performed by a human. This leads to a high demand for computer-aided or completely automated tagging approaches.

**String and Pattern Matching Approach.** You might for example be interested in finding and tagging all kinds of *places* in an unknown text file. Obviously, before being able to annotate a term referring to a *place* with additional information it is crucial to identify it correctly in the first place. This is a task most commonly performed by string matching or, more general, pattern matching algorithms.

Unfortunately, generic matching algorithms can encounter a large variety of problems: Spelling mistakes, OCR errors, typing variants and polysemy can inhibit the recognition process. To address these problems algorithms usually utilize external knowledge provided in lists of synonyms, flexion rules, grammars, spelling variants or common spelling mistakes for a given term. This knowledge helps to improve the overall classification performance.

The University of Würzburg hosts projects dealing with the preparation and presentation of ancient sources (Würzburg-University-Library, 2010). Ancient sources only have light spelling conventions and tend to follow a loose punction policy. For many terms, especially places or people, a broad variety of spelling variants exists. Furthermore, after digitalization the sources can contain many OCR-errors due to the sophisticated nature of the hand-writing at that time.

Performing a *Deep Tagging* on an ancient source is especially challenging due to its heterogeneous

Schöneberg H..
CONTEXT VECTOR CLASSIFICATION - Term Classification with Context Evaluation.
DOI: 10.5220/0003067403870391
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2010), pages 387-391
ISBN: 978-989-8425-28-7

appearance. Classification algorithms depending on string matching or pattern matching will therefore see their use limited in this scenario.

**Contextual Approach.** Not only the term itself, but its context, too, has proven to be a highly valuable source of information. According to (Miller and Charles, 1991), the exchangeability of two terms within a given context correlates to 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 term's 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 this information to create *Context Vectors* in a high-dimensional vector space to resolve polysemy. Apparently it is possible to gain information about a term by analyzing its context. The following example illustrates the idea of information extraction from a term's context:

**Example.** Imagine yourself passing by a group of people and overhearing a piece of conversation: *"Tomorrow I am going to fly to ..."*

Even though this sentence is not complete, it contains enough information for us to expect the missing word to be a *place*. In a conversation we would intuitively request the missing information by asking "Sorry, *where* are you going to?" and thereby express our expectation of a *place*. We classified the missing piece of information as *place* just by its context.

We expect the missing word to be a place, but our expectation ist not restricted to a specific place at all. This sentence would make perfect sense with a lot of terms, as long as they are instances of the class place: *Tomorrow I am going to fly to Berlin. Tomorrow I am going to fly to London.*

**Conclusion.** Consider two terms $s$ and $t$ as instances of class $x$. If $s$ and $t$ are exchangeable within a context $c$, then this context requires its related term to be of class $x$, regardless of its particular instantiation. (Miller and Charles, 1991) stated that semantic similarity correlates to contextual similarity.

Using the information contained in a given term's context allows two actions:

**Deduction of Knowledge.** Given the above example we expect the missing piece of information to be a place. If the speaker now replies with a word we have never heard so far, we would assume it to be a to us unknown place. That means, we classified a so far unknown term utilizing only the information within its context and acquired new knowledge.

**Verification of Knowledge.** If on the other hand the speaker replies with a term which, as far as we know, is not a place, we encounter a clash of knowledge: Maybe our data is correct and the speaker provided false information, maybe it's just contrary. In either case an erroneous piece of information would have been detected just by its context.

**Resolving Polysemy.** This is a special case of the before mentioned clash of knowledge. We might for example know for a fact, that a *crane* is a bird, but we could discover, that depending on its context this term could refer to a type of construction equipment, too.

We can suspect a term to be an instance of a certain class after evaluating its context, because as speakers of that particular language we understand the underlying rules of forming a sentence. With those rules in mind we can conclude, that only a few classes of terms would make actual sense in a given context.

Obviously, it is challenging to teach a computer to perform the same conclusions. Even with a sophisticated understanding of how to form a sentence in a given language, terms still have to be recognized in the first place, which brings us back to the recognition problems string and pattern matching algorithms can encounter (see page 1).

**Classification by Context.** The contextual information allows a transfer of knowledge to so far unknown words: If you can identify a context $c$, which demands its related term to be of class $x$, you could propose that whenever you happen to find another occurrence of $c$ within a source, its related term is an instance of class $x$, too. This leads to the following working assumption:

**Working assumption.** A classification algorithm can decide whether a given term is an instance of a class $x$ (e.g. $x = place$) by evaluating the context similarity.

**Statistical Context Analysis.** Given an arbitrary source $s$, let $n$ be the amount of *terms* within $s$. (Schütze, 1992) introduces a high-dimensional vector space with $n$ dimensions, one for each term in $s$. For any term $t$, its context can then be represented as a vector within this vector space, each dimension $d$ (which is a term, too) displaying the number $t$ and $d$ co-occurred throughout the source. The cosine angle (Baeza-Yates and Ribeiro-Neto, 1999) between two *Context Vectors* within this vector space measures the similarity of its terms co-occurrence-

patterns. Schütze suggests the usage of a fixed window size or sentence boundaries for the definition of co-occurrence.

**Part of Speech Analysis.** However, another source of information has not been taken into account so far: In most languages there are rules for forming a sentence. Not only does a valid sentence have to contain some integral parts (like subject, predicate, etc.), the language's grammar even implies a certain order of a sentence's components. By analyzing a sentence's sequence of terms we can gain additional information: Consider for example the expression *'the car'* - the occurrence of 'the' immediately before 'car' implies, that 'car' is a noun. On the other hand this implies, that -due to the language's grammar- many other parts of speech can *not* follow immediately after the occurrence of 'the', which of course will afflict term co-occurrence patterns. This information would be lost, if we discarded the term's position within a sentence. (Gauch et al., 1999) for example take into account a term's position within its context during co-occurrence-data analysis. Our approach utilizes a scoring mechanism which applies weighting factors to a term's co-occurrences based on their position within the context.

### 1.1 Performance Evaluation

**Independence.** The context evaluation approach is independent from the source's particular language, as it is an analysis of term co-occurrence patterns.

**Stability.** Imagine a source written in medieval German. This language follows only light spelling conventions, resulting in a large number of spelling variants for single terms. Regular string or pattern matching approaches will therefore have to depend on external knowledge to perform. However, even though a single term's spelling could vary in medieval German, the rules for forming a sentence were as strict as in any Germanic language today. That means, a place name had a specific context, regardless of its actual spelling variant. Of course even the terms forming the actual context surely had different spelling variants. Imagine a term referring to a place. Its context could contain prepositions like "to", "from", "in" etc. As these words occur a lot more frequent than the place they refer to, their spelling will be a lot more consistent throughout the source. The context evaluation approach is able to deal with weak orthography and spelling variants, taking advantage of a statistical evaluation of frequently used terms.

## 2 CONTEXT CLASSIFICATION ALGORITHM

Let $T^+$ be the set of **terms** containing relevant information.

1. We define a **class** as a set of terms $T_X$ with
$$T_X = \{t | \text{t is an instance of class } X, t \in T^+\}$$
2. Pick an arbitrary query element $q \in T_X$.
3. Evaluate the **context profile** $P_q$, which is the set of all **context items** $c_{q,i}$ for the $n$ occurrences of $q$, $1 \leq i \leq n$, with
$$P_q = \{c_q | c_q \text{ is context item for } q\}$$
and
$$c_{q,i} = \{t | t \in T^+ \text{ forming local context of hit } i\}$$
4. Each context item's component is assigned a score by the *scoring function* with
$$score : \Sigma^* \times \Sigma^* \rightarrow \mathbb{R}^+$$
A term's *overallScore* for a given *context profile* is the normalized sum of all scores:
$$overallScore(t, P_q) = \left( \sum_{i=1}^{|P_q|} score(t, c_i) \right) \left( \frac{1}{|P_q|} \right)$$
We assume a vector space in $\mathbb{R}^n$, with $n$ being the amount of terms in $T^+$. Each term $t \in T^+$ forms a dimension within the vector space. Given a query's $q$ **context profile** $P_q$ with $x$ terms and their respective *overallScore*, $P_q$ can be interpreted as a vector in this vector space. We use the standard vector model as discussed in (Baeza-Yates and Ribeiro-Neto, 1999). By interpreting a context as a vector in vector space we are now able to estimate two context profile's similarity by measuring the cosine angle between them.
5. Given a similarity threshold $\varepsilon$ with $0 \leq \varepsilon \leq 1$. For each context profile $P_r$ with a similarity exceeding the threshold $\varepsilon$ we propose:
$$q \in T_X \wedge similarity(P_q, P_r) \geq \varepsilon \quad \Longrightarrow \quad r \in T_X$$

## 3 OPTIMIZATIONS

Obviously, the classification quality is heavily impacted by the proper choice of the query term $q$ and its resulting context $P_q$. Consider the following example:

**Poor Representative.** Assume again, we are interested in finding all kinds of places throughout a given

source. According to our algorithm (see page 3) we choose a query term $q$ from the class of terms referring to places $T_{place}$. We choose a certain place $b$, which happens to host a famous regular sport event, but - aside from that - is fairly unknown otherwise. The query's context profile will most likely contain terms referring to the sport event. But obviously these attributes are not commonly shared for instances of the class $T_{place}$. Attributes, which on the other hand might be essential for identifying a place, could under circumstances not even occur within the context profile. The resulting context $P_b$ can therefore not reflect a *typical* context composition for an arbitrary place, even though it *is* a place. Each instance of the class $T_{place}$ could appear in slightly different context, resulting in a context with many terms relevant to only single instances, but not to the class. Clearly we need to find a way to identify the set of **significant terms** for a given class.

In order to extract the set of significant terms for a given class we cannot simply conjunct or intersect each instance's context: A conjunction would result in very large term sets, paying attributes, which are relevant to only few instances, too much attention in relation to the attributes relevant for the entire class. Intersection could on the other hand result in an empty term set, due to the overlapping nature of the context profiles.

**Majority Decision.** A majority decision is able to decide, which terms are relevant to a class rather than to particular instances. After choosing several instances of a class we calculate the most frequently used terms within their context profiles. After sorting the terms by frequency of occurrence we then define the top $i$ terms to be the **significant terms** for their class.

**Formal Description.** Let $T^+$ be the set of all relevant terms in our source.

1. Choose a class, e.g. $T_{place}$.
2. Pick $n$ terms from $T_{place}$ and calculate their context profiles.
3. Calculate $frequency(t)$ for each term $t \in T^+$ with
$$frequency(t) = \sum_{i=1}^{n} occurs(t, P_i)$$
and
$$occurs(t,c) = \begin{cases} 1 & \text{if } t \text{ occurs in context item } c \\ 0 & \text{otherwise} \end{cases}$$
4. Sort terms by their frequency and extract the most frequent $i$ items. We define the set of $i$ terms as $F^i_{place}$, the **i-significant** set of terms for the class *place*.

**Result.** We can use the majority decision defined above to determine the set $F^i_X$, the set of $i$ most significant terms for class $X$. Instead of comparing unknown context vectors with a single instance of our class $X$, we create a cluster of $n$ instances and extract the set of significant terms. With this we avoid using terms for comparison which might be relevant to only few instances of a given class. Each term's score is averaged from the overall scores. Each unknown context vector will then be compared with this new cluster context profile.

# 4 FUTURE WORK

**Deep Tagging.** The *Context Vector Classification* approach is designated to act as a support module for classic pattern matching algorithms for automated tagging. The Würzburg University Library is interested in processing (ancient) sources and annotating them according to the TEI-P5 standard (TEI-Consortium, 2007). Especially the detection of *events*, composed of actors, places and dates, is as important as difficult due to the syntactical challenges mentioned above. A workbench, which combines pattern matching algorithms with the *Context Vector Classification* approach, is under development with the goal of providing the user suggestions for the classification of terms.

**Reinforced Learning.** A learning module for the classification framework is currently under development. After a given term's classification has been proposed the user can approve or decline the decision. Based on the user's input a weight factor will be applied to each context vector's component. After several iterations the framework gains a specific *weight matrix* for a class of terms. This specialization allows an adaption to different contextual environments and improves the classification quality.

# 5 RESULTS

Up to this point only small, yet very promising tests of the classification quality have been conducted. Large-scale tests on corpora of different modern languages are currently under development.

## 5.1 Ancient Source

The *Context Vector Classification* approach was tested with an ancient German source, Merian's *Topographia Germaniae* (Merian, 1642) and (Merian,

Table 1: Most similar terms by cosine angle. Ancient German source (Merian, 1642).

| CLUSTER | HIGHEST SIMILARITY |
|---|---|
| places | mäyntz sachsen mayntz oesterreich vianden bamberg marpurg mümpelgart angefangen |
| names | friderich adolph johann georg otto albrecht wilhelm friederich ludwig heinrich |
| roles | bischoff könig abbt rath general hertzog thurn graff zeit käyser |

2010). Table 1 shows the most similar terms found for a given reference cluster, each cluster composed of five terms.

## 5.2 Modern Corpus

**Setup.** The following examples demonstrate the classification quality. The context vectors were created from a 3 million sentence corpus in German language (Leipzig-University, 1998).

**Places.** The test subject is a snippet of text containing 1658 terms, 26 of which relevant for classification as *place*. The contexter module examined a term's left, right and combined context with a window size of up to 6 terms. See (Baeza-Yates and Ribeiro-Neto, 1999) for the definitions of *precision*, *recall* and *f-value*. Table 2 shows the results.

Table 2: Cluster: Places. Context window size 6.

| CONTEXT | COS | PREC. | RECALL | F-VAL. |
|---|---|---|---|---|
| left | 0.9 | 0.73 | 0.85 | 0.79 |
| left | 0.95 | 0.95 | 0.77 | 0.85 |
| right | 0.9 | 0.07 | 0.81 | 0.13 |
| right | 0.95 | 0.26 | 0.62 | 0.36 |
| combined | 0.9 | 0.4 | 0.88 | 0.55 |
| combined | 0.95 | 0.95 | 0.81 | 0.88 |

**Names.** In this case the test subject is a snippet of text containing 1635 terms, 19 of which relevant for classification as *name*. The contexter module examined a term's left, right and combined context with a window size of up to 6 terms. See table 3 for results.

Table 3: Cluster: Names. Context window size 6.

| CONTEXT | COS | PREC. | RECALL | F-VAL. |
|---|---|---|---|---|
| left | 0.9 | 0.6 | 0.6 | 0.6 |
| left | 0.95 | 0.82 | 0.45 | 0.58 |
| right | 0.9 | 0.19 | 0.6 | 0.29 |
| right | 0.95 | 0.8 | 0.4 | 0.53 |
| combined | 0.9 | 0.5 | 0.7 | 0.58 |
| combined | 0.95 | 0.77 | 0.5 | 0.61 |

# 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.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. (1987). The vocabulary problem in human-system communication. *Commun. ACM*, 30(11):964–971.

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.

Leipzig-University (1998). German 3M corpus. http://corpora.informatik.uni-leipzig.de/.

Merian, M. d. A. (1642). *Topographia Germaniae*. Bärenreiter.

Merian, M. d. A. (2010). Topographiae Germaniae. http://de.wikisource.org/wiki/Topographia_Germaniae/.

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

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.

TEI-Consortium (2007). Guidelines for electronic text encoding and interchange. http://www.tei-c.org/release/doc/tei-p5-doc/en/html/.

Würzburg-University-Library (2010). Franconica online. http://franconica.uni-wuerzburg.de/Franconica/index.html/.