Learning Relations using Collocations

Gerhard Heyer, Martin Läuter, Uwe Quasthoff, Thomas Wittig, Christian Wolff
Leipzig University
Computer Science Institute, Natural Language Processing Department
Augustusplatz 10 / 11
D-04109 Leipzig
{heyer, laeuter, quasthoff, wittig, wolff}@informatik.uni-leipzig.de

Abstract

This paper describes the application of statistical analysis of large corpora to the problem of extracting semantic relations from unstructured text. We regard this approach as a viable method for generating input for the construction of ontologies as ontologies use well-defined semantic relations as building blocks (cf. van der Vet & Mars 1998). Starting from a short description of our corpora as well as our language analysis tools, we discuss in depth the automatic generation of collocation sets. We further give examples of different types of relations that may be found in collocation sets for arbitrary terms. The central question we deal with here is how to *postprocess* statistically generated collocation sets in order to extract *named relations*. We show that for different types of relations like *cohyponyms* or *instance-of*-relations, different extraction methods as well as additional sources of information can be applied to the basic collocation sets in order to verify the existence of a specific type of semantic relation for a given set of terms.

1 Analysis of Large Text Corpora

Corpus Linguistics is generally understood as a branch of computational linguistics dealing with large text corpora for the purpose of statistical processing of language data (cf. Armstrong 1993, Manning & Schütze 1999). With the availability of large text corpora and the success of robust corpus processing in the nineties, this approach has recently become increasingly popular among computational linguists (cf. Sinclair 1991, Svartvik 1992).

Since 1995 a German text corpus of more than 300 million words has been collected (cf. Quasthoff 1998B, Quasthoff & Wolff 2000), containing approx. 6 million different word forms in approx. 13 million sentences, which serves as input for the analysis methods described below. Similarly structured corpora have recently been set up for other European languages as well (English, French, Dutch), with more languages to follow in the near future (see table 1).

	German	*English*	*Dutch*	*French*
word tokens	300 M	250 M	22 M	15 M
sentences	13.4 M	13 M	1.5 M	860,000
word types	6 M	1.2 M	600,000	230,000

Table 1: Basic Characteristics of the Corpora

The basic goal of this corpus-based approach is to collect large amounts of textual data as input for semantic processing. Starting off from a rather simple data model tailored for large amounts of data and efficient processing using a relational data base system at storage level we employ a simple yet powerful technical infrastructure for processing texts to be included in the corpus. Beside basic procedures for text integration into the corpus various tools have been developed for post-processing linguistic data. Among them the automatic calculation of sentence-based word collocations stands out as an especially valuable tool for corpus-based language technology applications (see Quasthoff 1998A, Quasthoff & Wolff 2000). Additional, application oriented tools exist for search engine optimization as well as automatic document classification (see Heyer, Quasthoff & Wolff 2000). The corpora are available on the WWW (http://www. wortschatz. uni-leipzig.de) and may be used as a large online dictionary.

2 Collocations

The occurrence of two or more words within a well-defined unit of information (sentence, document) is called a collocation. For the selection of meaningful and significant collocations, an adequate collocation measure has to be defined. In the literature, quite a number of different collocation measures can be found; for an in-depth discussion of various collocation measures and their application cf. Smadja 1993, Lemnitzer 1998, Krenn 2000.

2.1 The Collocation Measure

In the following, our approach towards measuring the significance of the joint occurrence of two words *A* and *B* in a sentence is discussed. Let

- *a*, *b* be the number of sentences containing *A* and *B*,
- *k* be the number of sentences containing both *A* and *B*,
- *n* be the total number of sentences.

Our significance measure calculates the probability of joint occurrence of rare events. The results of this measure are quite similar to the well-known *log-likelihood*-measure (cf. Krenn 2000):

Let $x = ab/n$ and define:

$$\mathrm{sig}(A,B) = \frac{-\log\left(1-\mathrm{e}^{-x}\sum_{i=0}^{k-1}\frac{1}{i!}\cdot x^{i}\right)}{\log n}.$$

For $2x < k$, we get the following approximation which is much easier to calculate:

$$\mathrm{sig}(A,B) = (x - k\log x + \log k!) / \log n$$

In the case of next neighbor collocations we replace the definition of the above variables by the following. Instead of a *sentence* we consider pairs (*A*, *B*) of words which are next neighbors in this sentence. Hence, instead of one sentence of *n* words we have *n* - 1 pairs. For right neighbor collocations (*A*, *B*) let

- *a*, *b* be the number of pairs of type (*A*, *?*) and (*?*, B) resp.,
- *k* be the number of pairs (*A*, *B*),

n be the total number of pairs. This equals the total number of running words minus the number of sentences.

Given these variables, the significance measure is calculated as shown above. In general, this measure yields semantically acceptable collocation sets for values above an empirically determined positive threshold (see examples in section 3 below).

2.2 Properties of the Collocation Measure

In order to describe basic properties of this measure, we write sig(*n, k, a, b*) instead of sig(*A, B*) where *n, k, a,* and *b* are defined as above.

Simple co-occurance: *A* and *B* occur only once, and they occur together:

$$\mathrm{sig}(n,1,1,1) \to 1 \qquad (\text{for } n \to \infty).$$

Independence: *A* and *B* occur statistically independently with probabilities *p* and *q*:

$$\mathrm{sig}(n,npq,np,nq) \to 0 \qquad (\text{for } n \to \infty).$$

Additivity: The unification of the words *B* and *B'* just adds the corresponding significances. For $k/b \approx k'/b'$ we have

$$\mathrm{sig}(n,k,a,b) + \mathrm{sig}(n,k',a,b') \approx \mathrm{sig}(n,k+k',a,b+b')$$

Enlarging the corpus by a factor *m*:

$$\mathrm{sig}(mn, mk, ma, mb) = m\,\mathrm{sig}(n, k, a, b)$$

1.3 Finding Collocations

For calculating the collocation measure for any reasonable pairs we first count the joint occurrences of each pair. This problem is complex both in time and storage. Nevertheless, we managed to calculate the collocation measure for any pair with total frequency of at least 3 for each component. Our approach is based on extensible ternary search trees (cf. Bentley & Sedgewick 1998) where a count can be associated to a pair of word numbers. The memory overhead from the original implementation could be reduced by allocating the space for chunks of 100,000 nodes at once. Even when using this technique on a large memory computer more than one run through the corpus may be necessary, taking care that every pair is only counted once. The resulting word pairs above a threshold significance are put into a database where they can be accessed and grouped in many different ways. As collocations are calculated for different language corpora, our examples will be taken from the English as well as the German database.

1.4 Visualization of Collocations

Beside textual output of collocation sets, visualizing them as graphs is an additional type of representation: We choose a word and arrange its collocates in the plane so that collocations between collocates are taken into account. This results in graphs that show homogeneity where words are interconnected and they show separation where collocates have little in common. Linguistically speaking, polysemy is made visible (see fig. 1 below).

Technically speaking, we use *simulated annealing* to position the words (see Davidson & Harel 1996). Line thickness represents the significance of the collocation. Of course, all words in the graph are linked to the central word, the rest of the picture is automatically computed, but represents semantic connectedness surprisingly well. Unfortunately the relations between the words are just presented, but not yet named. Fig. 1 shows the collocation graph for *space*. Three different meaning contexts can be recognized in the graph:

- real estate,
- computer hardware, and
- astronautics.

The connection between *address* and *memory* results from the fact that address is another polysemous concept.

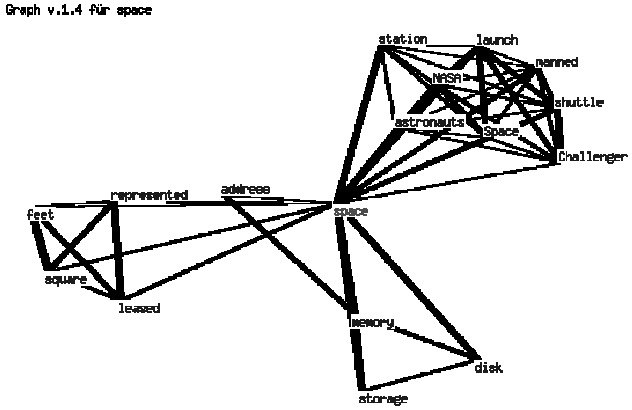


Fig. 1: Collocation Graph for space

3 Relations Represented by Collocations

If we fix one word and look at its set of collocates, then some semantic relations appear more often than others. The following example shows the most significant collocations for *king* ordered by significance:

queen (90), mackerel (83), hill (49), Milken (47), royal (44), monarch (33), King (30), crowned (30), migratory (30), rook (29), throne (29), Jordanian (26), junk-bond (26), Hussein (25), Saudi (25), monarchy (25), crab (23), Jordan (22), Lekhanya (21), Prince (21), Michael (20), Jordan's (19), palace (19), undisputed (18), Elvis (17), Shah (17), deposed (17), Panchayat (16), Zahir (16), fishery (16), former (16), junk (16), constitution (15), exiled (15), Bhattarai (14), Presley (14), Queen (14), crown (14), dethroned (14), him (14), Arab (13), Moshoeshoe (13), himself (13), pawns (13), reigning (13), Fahd (12), Nepali (12), Rome (12), Saddam (12), once (12), pawn (12), prince (12), reign (12), [...] *government (10)* [...]

The following types of relations can be identified:

- Cohyponymy (e. g. *Shah*, *queen*, *rook*, *pawn*),
- top-level syntactic relations, which translate to semantic 'actor-verb' and often used properties of a noun (*reign*; *royal*, *crowned*, *dethroned*),
- instance-of (*Fahd*, *Hussein*, *Moshoeshoe*),
- special relations given by multiwords (*A* prep/det/conj *B*, e. g. *king of Jordan*), and
- unstructured set of words describing some subject area, e. g. *constitution*, *government*.

Note that synonymy rarely occurs in the lists. The relations may be classified according to the properties symmetry, anti-symmetry, and transitivity.

3.1 Symmetric Relations

Let us call a relation *r* symmetric if *r*(*A*, *B*) always implies *r*(*B*, *A*). Examples of symmetric relations are

- synonymy,
- cohyponomy (or similarity),
- elements of a certain subject area, and
- relations of unknown type.

Usually, sentence collocations express symmetric relations.

3.2 Anti-symmetric Relations

Let us call a relation *r* anti-symmetric if *r*(*A*, *B*) never implies *r*(*B*, *A*). Examples of anti-symmetric relations are

- hyponymy and
- relations between properties and its owners like action and actor or class and instance.

Usually, next neighbor collocations of two words express anti-symmetric relations. In the case of next neighbor collocations consisting of more than two words (like *A prep/det/conj B* e. g. *Samson and Delilah*), the relation might be symmetric, for instance in the case of conjunctions like *and* or *or* (cf. Läuter & Quasthoff 1999).

3.3 Transitivity

Transitivity of a relation means that *r*(*A*, *B*) and *r*(*B*, *C*) always implies *r*(*A*, *C*). In general, a relation found experimentally will not be transitive, of course. But there may be a part where transitivity holds.

Some of the most prominent transitive relations are the *cohyponymy*, *hyponymy*, *synonymy*, and *is-a* relations. Note that our graphical representation mainly shows transitive relations per construction. This kind of relation is also able to give further results in the combination procedures described below.

4 Other Sources for Relations

While we may intellectually identify types of semantic relations in collocations sets, additional information and / or analysis is needed for *automatically* naming these relations. In the following, we give different examples for such complementary information.

4.1 Pattern Based Relations

Simple pattern-based relations can be extracted from text if knowledge about information categories like proper names is used as input. As our corpora include several large lists of classified terms like names of professions and last names, extraction rules may be defined:

i. Extraction of first names:
   A pattern like *(profession) ? (last name)* implies (with high probability) that the unknown category *?* is in fact a *first name*. Examples are
   actress *Julia* Roberts
   hockey hero *Wayne* Gretzky
   Senator *Jesse* Helms

ii. Extraction of *instance-of*-relations given the class name: The pattern *(class name) like ?* implies (with high probability) that the unknown category *?* is in fact a *instance name*. Examples are:
   metals like *nickel*, arsenic and lead
   rivers like the *Ganges*
   newspapers like *Pravda*

The applicability of patterns like these may heavily depend on language characteristics like preposition usage. This type of extraction method is simple and well known; in our approach it is combined with collocation analysis, thus yielding better results both in quality and in quantity (see section 5).

4.2 Compounds

German compounds consist of two (or more) words glued together by varying mechanisms. The head word (coming second) is further determined by the first part of the compound (modifier), which may originally be an adjective, another noun or a verb stem. In almost all cases a semantic relation between both parts and the compound can be found. In section 5.3 we show how the combination of compound segmentation with collocation analysis can be used for identifying named relations in compounds.

4.3 Feature Vectors Given by Collocations and Clustering

To investigate the meaning of a word *A*, its contexts in the texts have to be examined because they reflect the use of *A*. If two words *A* and *B* have similar contexts, that is, they are alike in their use, this indicates that there is a semantic relation between *A* and *B* of some kind.

A kind of *average context* for every word *A* is formed by all collocations for *A* with a significance above a certain threshold.

This average context of *A* is transferred into a feature vector of *A* using all words as features as usual. This results in sparse vectors used for description. The feature vector of word *A* is indeed a description of the meaning of *A*, because the most important words of the contexts of *A* are included.

Clustering of feature vectors can be used to investigate the relations between a group of similar words and to figure out whether or not all the relations are of the same kind.

The following HACM algorithm has an additional natural reason to stop. It works bottom up like this:

- All words are treated as (basic) items. Each item has a description (feature vector).
- In each step of the clustering process the two items *A* and *B* with the most similar description vectors are searched and fitted together to create a new complex item *C* combining the words in *A* and *B*. The scalar product is used for determining similarity between vectors.
  Each step of the clustering algorithm reduces the number of items by one.
- The feature vector for *C* is constructed from the feature vectors of *A* and *B*. Therefore we calculate a combined significance for C with respect to all words X_i as follows:

$$\text{sig}(C, X_i) = \frac{n_a}{n_a + n_b}\text{sig}(C, X_i) + \frac{n_b}{n_a + n_b}\text{sig}(C, X_i)$$

for all i, $1 \le i \le n$ with
n total number of words in the corpus,
n_a number of words combined in item A, and
n_b number of words combined in item B.

- The algorithm stops if only one item is left or if all remaining feature vectors are orthogonal. This results usually in a very natural clustering if the threshold for constructing the feature vectors is suitably chosen.

A cluster of words with probably the *same semantic relation* between each of them can be found in the analysis tree by comparing the similarity between items inside the items *A* and *B* (if these items are complex) with the calculated similarity between *A* and *B*, when fitting them together to *C*. If there is a large difference between them, this is an indication for a different relation between words combined in item *A* and words combined in item *B*. In the appendix, some examples for this type of semantic clustering are given.

Symmetric clustering
If we assume that a cluster represents a semantic relation, the cluster should represent the possible symmetry and transitivity of the underlying semantic relation.
Symmetry and transitivity ensure that the terms to be clustered will themselves be responsible for the clustering. This in turn implies that the terms found in the cluster will also be found in the feature vector in prominent positions.
In example 1 (Appendix) the clustering result for *January* is shown. In the first column we find the terms to be clustered, on the right hand side there are the components of the feature vectors ordered by significance.
The clustered items both appear together and share a certain aspect. The names of the months or weekdays as names for periods of time cluster together, just because they are collocates with one another. The same can be shown to be true for teammates, metals, colors or fruit.

Anti-symmetric clustering
For anti-symmetric relations the situation is different. Again the elements of the original set to be clustered share a certain aspect, but this aspect is described by a distinct set of words. Presumably this second set of words will also cluster. Moreover, it will use the original set as clustering terms.
This is shown in example 2 (Appendix). Here we show that the set given by *Präsident, Vorsitzender, Vorsitzende, Sprecher, Sprecherin* properly clusters using words like *sagte, erklärte, teilte* (German *verbs of utterance*).
Conversely, in example 3 (Appendix) we find the set *verwies, mitteilte, meinte, bestätigte, betonte* properly clusters using terms from the above cluster.

4.4 Homogeneous Relations: Iterating the Collocation Process

The extraction of collocation sets from plain text can be viewed as some kind of information condensation. This process can be iterated if collocation sets themselves are subjected to the collocation analysis again and again. We might expect that some of the collocational relations are strengthened while others will vanish from the iterated sets of collocations which we will call higher order collocations. We describe two experiments for the iteration process: Instead of plain text we start with collocation sets, using sentence collocations for experiment 1 and next neighbor collocations for experiment 2. In the case of a *symmetric relation* we observe a strengthening while iterating sentence collocations. In the case of an *anti-symmetric relation* we observe the same when iterating next neighbor collocations.

Experiment 1: Iterating Sentence Collocations
The production of collocations is applied to sets of sentence collocations instead of sentences. E.g., the collection of 500,000 sentence collocations has the following ‘sentence‘ (collocation set) for *Hemd* (*shirt*): *Hemd Krawatte Hose weißes Anzug weißem Jeans trägt trug bekleidet weißen Jacke schwarze Jackett schwarzen Weste kariertes Schlips Mann*

Example for iterated sentence collocations of *Eisen* (*iron*):
Original collocations: *Stahl, heißes, heiße, Kupfer, Mangan, alten, Feuer, Zink, Holz, Marmor*
Iterated collocations: *Kupfer, Stahl, Zink, Aluminium, Magnesium, Mangan, Nickel, Blei, Zinn, Gold*
As expected, the iterated collocation set only contains cohyponyms.

Experiment 2: Iterating Next Neighbor Collocations
In this experiment, the production of collocations is applied to sets of *next neighbor* collocations instead of sentences. The collection of 250,000 next neighbor collocations has the following two ‘sentences‘ for *Hemd* (*shirt*):
weißes weißem weißen blaues kariertes kariertem offenem aufs karierten gestreiftes letztes [...] (left neighbors)

näher bekleidet ausgezogen spannt trägt aufknöpft ausgeplündert auszieht wechseln aufgeknöpft ausziehen [...] (right neighbors)
Example for iterated neighbor collocations of *Auto* (*car*):
Original collocations: *fahren, Wagen, prallte, Fahrer, seinem, fuhr, fährt, Polizei, erfaßt, gefahren*

Iterated collocations: *Wagen, Lastwagen, Fahrzeug, Autos, Personenwagen, Bus, Zug, Haus, Lkw, Pkw*
Example for iterated neighbor collocations of *erklärte* (*explained*):
Original collocations: *Sprecher, werde, gestern, seien, Wir, bereit, wolle, Vorsitzende, Anfrage, Präsident*
Iterated collocations: *sagte, betonte, sprach, kündigte, wies, nannte, warnte, bekräftigte, meinte, kritisierte*

Both, experiment 1 and experiment 2 result in collocation sets carrying a homogeneous semantic relation.

5 Combining Non-contradictory Partial Results

In section 3 we have given evidence that collocation sets contain various types of semantic relations without explicitly naming them while section 4 has introduced a

number of methods for relation extraction. This section shows different ways of *combining* results of these extraction approaches. The results of these combination give more and / or better results.

5.1 Identical Results

Two or more of the above algorithms may suggest a certain relation between two words, for instance, cohyponymy.
Example: If both the second order collocations introduced in section 4.4, and clustering by feature vectors (section 4.3) independently yield similar sets of words as a result, this may be taken as an indication of cohyponymy between the words, e. g. *sagte, betonte, kündigte, wies, nannte, warnte, bekräftigte, meinte* […] (German *verbs of utterance*).

5.2 Supporting Second Results

In the second combination type a *known* relation given by one method of extraction is verified by an identical but *unnamed* second result as follows:
Result 1: There is certain relation *r* between *A* and *B*
Result 2: There is some strong (but unknown) relation between A and B (e. g. given by a collocation set)
Conclusion: Result 1 holds with more evidence.
One can use this support of orthogonal tests in many ways: Without knowing anything about deeper language structure or parsing we can filter out verbs just by testing if a string accepts at least two of the endings *–(e)s, -ing* and *–ed/t*. The recall is remarkably high. In German we tested only one mechanism of noun formation from a verb and got 70% of all verbs with a precision of 83%.
Word formation mechanisms can be explored further. In German compound nouns are joint together to form one word. There are several (highly irregular) patterns of gluing letters between the words. Testing all available word tokens whether they *could* be the compound of two stemmed words from word lists of 93,000 current nouns reveals just under a million compounds in their stemmed form. Here stemming accuracy is supported by the existence of both compounds in the basic list. When eliminating a hundred words which are prone to generate wrong separations this algorithm achieves an accuracy of 90%.
Example:
Result 2: The German compound *Entschädigungsgesetz* can be divided into *Gesetz* and *Entschädigung* with an unknown relation.
Result 1 is given by the four word next neighbor collocation *Gesetz über die Entschädigung*. Similarly *Stundenkilometer* is analyzed as *Kilometer pro Stunde.*
In these examples, result 1 is not enough because there are collocations like *Woche auf dem Tisch* which do not describe a meaningful semantic relation.

5.3 Combining Three Results

Result 1: There is relation *r* between *A* and *B*
Result 2: *B* is similar to *B'* (cohyponymy)
Result 3: There is some strong but unknown relation between *A* and *B'*
Conclusion: There is a relation *r* between *A* and *B'*
Example: As result 1 we might know that *Schwanz* (*tail*) is part of *Pferd* (*horse*). Similar terms to *Pferd* are both *Kuh* (*cow*) and *Hund* (*dog*) (result 2). Both of them have the term *Schwanz* in their set of significant collocations (result 3). Hence we might correctly conjecture that both *Kuh* and *Hund* have a *tail* (*Schwanz*) as part of their body. In contrast, *Reiter* (*rider*) is a strong collocation to *Pferd* and might (incorrectly) be conjectured to be another similar concept, but *Reiter* is no collocation with respect to *Schwanz*. Hence, the absence of result 3 prevents us from making an incorrect conclusion.

5.4 Similarity Used to Infer a Strong Property

Let us call an property *p* important, if it is preserved under similarity. This strong feature can be used as follows:
Result 1: *A* has a certain important property *p*
Result 2: *B* is similar to *A* (i. e., *B* is a cohyponym of *A*)
Conclusion: *B* has the same property *p*
Example: We consider *A* and *B* as similar if they are in the set of right neighbor collocations of *Hafenstadt* (*port town*) (result 2). If we know that *Hafenstadt* is a property of its typical right neighbors (result 1) we may infer this property for more then 200 cities like *Split, Sidon, Durban, Kismayo, Tyrus, Vlora, Karachi, Durres,* […].

5.5 Subject Area Inferred from Collocation Sets

Result 1: *A, B, C, ...* are collocates of a certain term.
Result 2: Some of them belong to a certain subject area.
Conclusion: All of them belong to this subject area.
Example: Consider the following top entries in the collocation set of *carcinoma*: *patients, cell, squamous, radiotherapy, lung, thyroid, treated, hepatocellular, metastases, adenocarcinoma, cervix, irradiation, breast, treatment, CT, therapy, renal, cases, bladder, cervical, tumor, cancer, metastatic, radiation, uterine, ovarian, chemotherapy,* […]
If we know that some of them belong to the subject area *Medicine*, we can add this subject area to the other members of the collocation set as well.

6 Conclusion

In this paper, we described different approaches for the extraction of named semantic relations from large text corpora. The types of relations are compatible with relations typically used for constructing ontologies (cf. Chandrasekaran 1999:22). The combination of different types of input information as well as the application of robust statistical analysis methods guarantees that this approach may be applied to texts from arbitrary domains and different languages. Especially, our results may be used for the automatic generation of semantic relations in order to fill and expand ontology hierarchies.

7 References

Armstrong, S. (ed.) (1993). Using Large Corpora. Computational Linguistics 19(1/2) (1993) [Special Issue on Corpus Processing, repr. MIT Press 1994].
Bentley, J.; Sedgewick, R. (1998). "Ternary Search Trees." In: Dr. Dobbs Journal, April 1998.

Chandrasekaran, B. et al. (1999). “What are Ontologies, and Why Do We Need Them?” In: Intelligent Systems 14(1) (1999), 20-26.

Davidson, R., Harel, D. (1996). “Drawing Graphs Nicely Using Simulated Annealing.” In: ACM Transactions on Graphics 15(4), 301-331.

Francis, W.; Kucera, H. (1982). Frequency Analysis of English Language. Boston: Houghton Mifflin.

Heyer, G.; Quasthoff, U.; Wolff, Ch. (2000). “Aiding Web Searches by Statistical Classification Tools.“ In: Knorz, G.; Kuhlen, R. (eds.) (2000). Informationskompetenz - Basiskompetenz in der Informationsgesellschaft. Proc. 7. Intern. Symposium f. Informationswissenschaft, ISI 2000, Darmstadt. Konstanz: UVK, 163-177.

Krenn, B. (2000). “Distributional and Linguistic Implications of Collocation Identification.” In: Proc. Collocations Workshop, DGfS Conference, Marburg, March 2000.

Läuter, M., Quasthoff, U. (1999). “Kollokationen und semantisches Clustering.” In: Gippert, J. (ed.) (1999). Multilinguale Corpora. Codierung, Strukturierung, Analyse. Proc. 11. GLDV-Jahrestagung. Prague: Enigma Corporation, 34-41.

Lemnitzer, L. (1998). “Komplexe lexikalische Einheiten in Text und Lexikon.” In: Heyer, G.; Wolff, Ch. (eds.). Linguistik und neue Medien. Wiesbaden: Dt. Universitätsverlag, 85-91.

Manning, Ch. D.; Schütze, H. (1999). Foundations of Statistical Language Processing. Cambridge/MA, London: The MIT Press.

Quasthoff, U. (1998A). “Tools for Automatic Lexicon Maintenance: Acquisition, Error Correction, and the Generation of Missing Values.“ In: Proc. First International Conference on Language Resources & Evaluation [LREC], Granada, May 1998, Vol. II, 853-856.

Quasthoff, U. (1998B). “Projekt der deutsche Wortschatz.” In: Heyer, G., Wolff, Ch. (eds.). Linguistik und neue Medien. Wiesbaden: Dt. Universitätsverlag, 93-99.

Quasthoff, U.; Wolff, Ch. (2000). “An Infrastructure for Corpus-Based Monolingual Dictionaries.” In: Proc. LREC-2000. Second International Conference On Language Resources and Evaluation. Athens, May/June 2000, Vol. I, 241-246.

Sinclair, J. (1991). Corpus Concordance Collocation. Oxford: Oxford University Press.

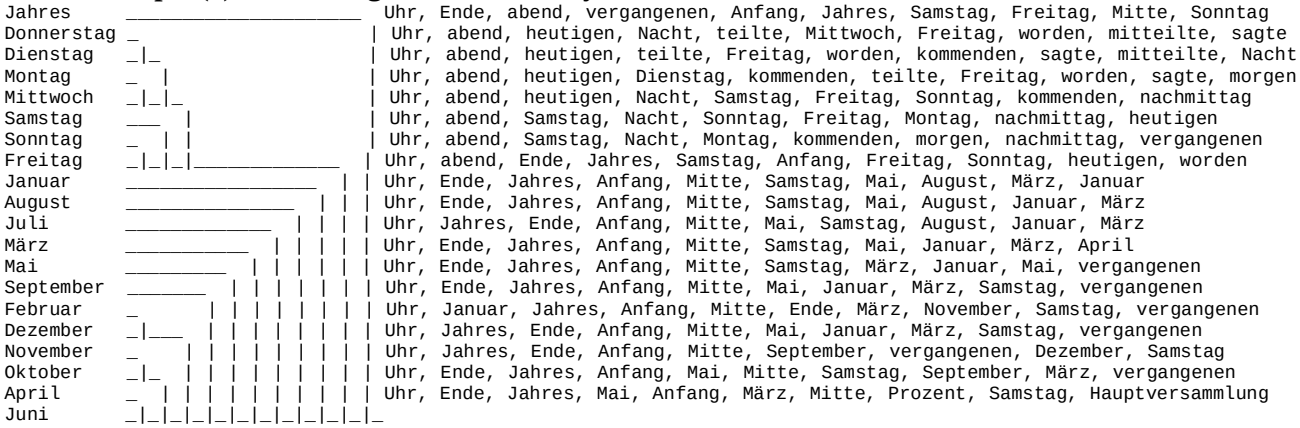
Smadja F. (1993). “Retrieving Collocations from Text: Xtract.” In: Computational Linguistics 19(1) (1993), 143-177.

Svartvik, J. (ed.) (1992). Directions in Corpus Linguistics: Proc. Nobel Symposium 82, Stockholm, 4-8 August 1991. Berlin: Mouton de Gruyter [=Trends in Linguistics 65].

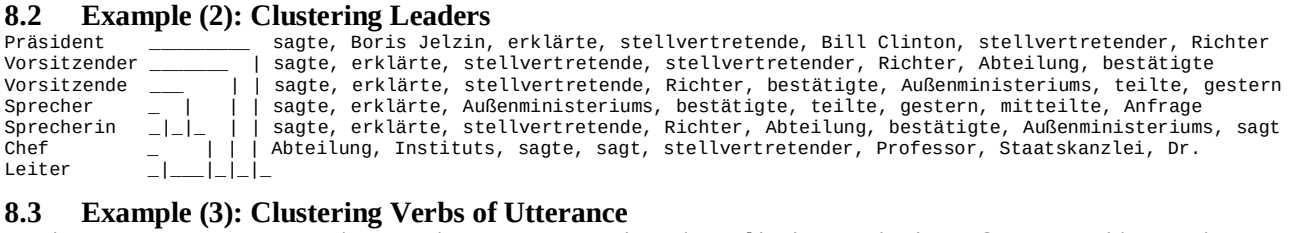
van der Vet, P. E.; Mars, N. J. I. (1998). “Bottom-Up Construction of Ontologies.” In: IEEE Transactions on Knowledge and Data Engineering 10(4) (1998), 513-526.

8 Appendix: Clustering Examples

8.1 Example (1): Clustering Months and Days



8.2 Example (2): Clustering Leaders



8.3 Example (3): Clustering Verbs of Utterance