*Journal of Universal Computer Science, vol. 9, no. 6 (2003), 530-541*
*submitted: 27/1/03, accepted: 17/3/03, appeared: 28/6/03 © J.UCS*

# Automatic Discovery and Aggregation of Compound Names for the Use in Knowledge Representations

Christian Biemann
(University of Leipzig, Germany
biem@informatik.uni-leipzig.de)

Uwe Quasthoff
(University of Leipzig, Germany
quasthoff@informatik.uni-leipzig.de)

Karsten Böhm
(University of Leipzig, Germany
boehm@texttech.de)

Christian Wolff
(Chemnitz University of Technology
christian.wolff@informatik.tu-chemnitz.de)

**Abstract:** Automatic acquisition of information structures like Topic Maps or semantic networks from large document collections is an important issue in knowledge management. An inherent problem with automatic approaches is the treatment of multiword terms as single semantic entities. Taking company names as an example, we present a method for learning multiword terms from large text corpora exploiting their internal structure. Through the iteration of a search step and a verification step the single words typically forming company names are learnt. These name elements are used for recognizing compounds in order to use them for further processing. We give some evaluation of experiments on company name extraction and discuss some applications.

**Key Words:** Corpora, Semantic Relations, Topic Maps, Text Mining, Knowledge Management, Named Entity Extraction
**Categories:** H.3.3, H.5.3, I.2.6, I.2.7, I.7

## 1 Introduction

Since 1994 we have been setting up an infrastructure for processing and analysing electronic text corpora [see Quasthoff and Wolff 2000]. The underlying corpora are accessible on the web (see http://www.texttech.de, http://wortschatz.uni-leipzig.de) and comprise a very large online corpus for German, as well as for other languages like English or Dutch. The core of this infrastructure are statistical algorithms for collocation analysis which compute significant collocations for all word types at sentence level or requiring immediate neighbourhood [see Heyer et al. 2001]. Using this framework we have developed a solution for the automatic creation of Topic Maps that can be used for the automatic structuring of large document collections in the area of Knowledge Management.

Over the last two years we have been using text mining technologies for the construction of structural representations (Topic Maps) from raw textual data to be

used in Knowledge Management solutions [see Boehm et al. 2002]. While working in productive environments with customer data we discovered some immanent flaws in the resulting structures that forced us to investigate additional methods on top of the existing technology to improve the overall quality. In this paper we introduce a method for extracting multiword terms as a preliminary step. Those gazetteers (e.g. of company names, product names, person names) can be used for

1. recognizing named entities (e.g. product and company names) that consist of multiple words
2. aggregating the different variants occurring in the text into a single entity (e.g. topic in the field of Topic Maps).
3. assign types to previously unknown entities

In [Quasthoff et al. 2002] an approach is presented that learns a large set of person names from a small initial knowledge set and a few rules of thumb, using a bootstrapping process. Here we use a similar algorithm for the more complicated task of extracting company names from an unannotated large corpus of German. In this paper we present a solution for extracting multiword named entities from large unstructured corpora and show how existing applications in text mining and knowledge management may be enhanced by this approach.

## 2 Company Names

### 2.1 Motivation

As an initial approach for the automatic detection and aggregation of compound names we focused on company names. Our motivation for the choice of this subclass of compound names arises from a number of reasons.

The first motivation was the importance of these types of concepts in the area of knowledge structuring in unstructured documents. While working in these fields we constantly noted that it was often vital for the users of automatic generated knowledge structures to find well-known entities, like companies modelled within the network of concepts (e.g. a Topic Map). Interestingly the names of companies found in ordinary documents often did not match the regular name of the company exactly, where abbreviated or shortened. While the human reader of such documents can easily establish the connection to a single concept representing the company, this step is much harder to accomplish for a text mining process, which has no prior knowledge of the company name parts that can be omitted without blurring the semantics in the context of the document.

On the other hand company names are compound names that are well known in the sense that they are listed and maintained in commercial registers, yellow pages and similar type of lists that should be made available to the public. Recognition of company names should be as simple as comparing these lists to the concepts found within the document collection. The fact that entries in these registers and lists are often not easily accessible (fees, licenses, different file formats), maintained by different organizations at different levels (national, international, governmental, communal etc.) and that the entries change frequently impose severe problems on

such a current list of company names that can be used for reliable named entity recognition.

This situation leads to the surprising situation that it still is a tricky problem to extract company names as multiwords from unannotated text sources and that it is even hard to know how they can be appropriately aggregated.

### 2.2 Observations

Looking at company names in German texts, it may be observed that complete company names follow two types of regularities: First, they follow certain *structural* patterns. At second, many elements (i.e. words) of company names reappear in different company names. Moreover, they reappear in the same or a similar position. For instance, *GmbH* and *Ltd.* can be found near the end of a company name in most cases. We distinguish three distinct categories in our analysis of company names:

1. Initial abbreviations (ABBR),
2. a field containing name parts (NAME) possibly connected by conjunctions, and
3. abbreviations of the legal form (KIND).

Table 1 shows some examples for this analysis (all examples are taken from our German reference corpus).

| Abbreviation (ABBR) | Name parts (NAME) | Legal Form (KIND) |
|---|---|---|
| A & W | Elektrogeräte | GmbH & Co. KG |
| A. | Baumgarten | GmbH |
| | Hagedorn | GmbH |
| | Institut für Angewandte Kreativität | |
| DASAG | | GmbH |
| | Japan Steel Works | Ltd. |
| K.F.C. | Germany | Inc. |
| LABSCO | Laboratory Supply Company | GmbH & Co. KG |

*Table 1: Fields of Company names*

Further analysis shows that the abbreviation field contains sequences of capitals, full stops (FS) and conjunctions (CONJ), while the name parts consist of generic company terms, person names, locations and connectives (CONN), and the legal form field (KIND) is an enumeration of legal form abbreviations, possibly also containing Conjunctions and punctuation marks.

This initial analysis can be formalised by defining patterns for those categories that match company names, a quite well-known approach in named entity extraction [see Greenwood and Gaizauskas 2003]. It is possible to describe the structure of most company names with a straightforward regular expression like

```
(ABBR(FS|CONJ)?)* (NAME|CONN)* (KIND(FS|CONJ)?)*
```

For our computation we impose some restrictions on the Kleene star (*) semantics, as we do not want the empty expression to be regarded a company name. In practice, we expand the regular expression to a larger set of patterns like ABBR NAME KIND („A. Baumgarten GmbH“) or ABBR NAME NAME NAME KIND CONJ KIND FS KIND („LABSCO Laboratory Supply Company GmbH & CO. KG“), just to name a few.

# 3 Algorithm

Our assumptions concerning the regularity of named entities motivate the following iterative approach: The algorithm starts with an initial set of pattern rules and initial elements of the different structural categories ABBR, NAME, and KIND. A large corpus of more than 15 million sentences [see Quasthoff and Wolff 2000] is used for both the identification of candidates for new elements of the goal classes as well as for the verification of these candidates. Newly learnt elements are used to identify further candidates in the next step, and the alternation of search step and verification step is iterated until no more elements are found. Similar approaches can be found in [Yangarber et al. 2002] for the extraction of generalized names (such as illnesses and medicaments) and in [Duclaye et al. 2002], which uses the web to find semantic relations.

## 3.1 Building Pattern Rules from Patterns

Besides finding and identifying additional candidates for company name parts, our approach is also capable of extracting additional patterns for company names: The initial set of extraction rules. In this section we describe how to build extraction rules from the patterns we gave in the previous paragraph. The pattern rules serve as an instrument to classify previously unknown words as belonging to one of the goal classes. From a pattern like ABBR NAME KIND we form the rules given in the following table.

| | | |
|---|---|---|
| _CAP* NAME KIND | → | ABBR |
| ABBR _UC* KIND | → | NAME |
| ABBR NAME _MIX* | → | KIND |

*Table 2: Rules constructed from a pattern* [1].

The first pattern rule in table 2 should be read in the following way: If there is a sequence of a capitalized word, a word belonging to the NAME class and a word belonging to the KIND class, then the first word is likely to belong to ABBR. Rules constructed that way are obviously massively overgeneralized. It is not difficult to

[1] _CAP (capitalized), _UC (upper case) and _MIX (mix of upper case and lower case letters) are features of the unknown word, the position marked by the * is classified as belonging to the class in the right side after the "⟶".

find examples where those rules predict wrong classes. However, the verification step below ensures that these "false friends" are eliminated with high accuracy.

### 3.2 Learning Elements by Search and Verification

Since the intellectual formulation of generally applicable rules is a time-consuming task that can always fail on "real-world" data, we use a different approach that is tolerant to single misclassifications of the rules. Using the algorithm described below (fig. 1), we judge the quality of the match on some sample sentences.

```
Initialise pattern rules
Let unused elements ::== initial set of elements
   belonging to some category

Loop:
 For each unused element
   Find candidates for new elements by the □□□□□□□□□□□□
        For each candidate
          Do the □□□□□□□□□□□□□□□□□□
          Add accepted candidates to new unused elements
 Output new unused elements
 Unused elements = new unused elements
```

*Figure 1: Search and Verification algorithm*

Both the search step and the verification step use the corpus. In the search step, the pattern rules are applied to sentences containing the actual unused element. This yields a list of candidates together with their guessed categories. In the verification step, the corpus is searched for sentences containing a candidate. We count the number of occurrences of this candidate where the pattern rules predict its guessed category. If the ratio of positive classifications against the total number of occurrences is above a certain threshold, the candidate is accepted.

Note that we use *sentences* in the search step rather than only text windows of some words containing the unused element. This is due to a locality principle which states that terms belonging to the same class often occur together, e.g. in business news often several companies appear in a row.

To exemplify the algorithm, suppose we know that "Film" (movie) belongs to the NAME class, "AG" and "GmbH" belong to the KIND class. The rule set in our small example is built from the pattern "NAME NAME KIND". In the search step, German sentences containing the following fragments are extracted:

| | |
|---|---|
| Die *CineMedia* Film AG übernahm | die *Odeon* Film AG mit |
| der TaunusFilm-Produktions GmbH in | der Bavaria Filmverleih- und Produktions GmbH den |
| ihre Tochtergesellschaft TaunusFilm-Produktions GmbH | die Lunaris Film mit |
| darunter ein Film über | zu jedem Film interessante |
| die *Senator* Film AG über | zukunftsweisenden Film "Jurassic Park" |
| die *Lunaris* Film GmbH | erfolgreichsten Film der |
| der *Odeon* Film AG. | dem Film eine Hauptrolle |

*Table 3: Extraction Examples*

Items marked in italics are candidates for items belonging to NAME, i.e. “CineMedia”, “Senator”, “Lunaris” and “Odeon”. The underlined sequences are proper company names that are not recognized by our example pattern. Note that the patterns help to distinguish between the readings of “Film” as the noun “movie” and as a part of a company name.

In the verification step, sentences containing e.g. “CineMedia” are requested from the corpus. The candidates do not get their guessed class assigned to them. The pattern rules are applied and the number cases where “CineMedia” is again classified as NAME is compared to the sum of all appearances of “CineMedia”. In our example, “CineMedia”, “Lunaris” and “Odeon” are verified and can be used for the search step in the next iteration. The reason for rejecting “Senator” is the frequent reading as “member of senate”.

### 3.3 Extraction of Terms

Through the iteration, a large number of words that are likely to appear in company names are learnt. These results are used to build gazetteers of company names. When extracting terms from unannotated text, it is crucial to recognize term boundaries correctly. In many cases an article to the left and a small initial word to the right delimit company names in German. Another possibility is to look for the longest match of the regular expression.

In practice, we define a delimiter category (DELIM_L and DELIM_R for left and right delimiters) and build term extraction patterns by adding the delimiters to the patterns, i.e. DELIM_L ABBR NAME KIND DELIM_R.

Together with the elements learnt in the iteration, these patterns form a powerful instrument to extract company names from unannotated texts of arbitrary length by a two-pass method:

1. The text is annotated by means of the elements and by flat features like _CAP, _UC, _MIX and _LC (lower case). In the first pass the pattern rules are applied and the verification step is performed on the large background corpus. Newly learnt elements are incorporated in the annotation.
2. The second pass extracts full terms by applying the extraction patterns on the annotation sequence.

### 3.4 Related Work

The algorithm implements bootstrapping by using previously learnt items to find and verify new ones, using its own output as input. Related methods can be found in [Riloff and Jones 1999], [Collins and Singer 1999] and [Duclaye et al. 1999]. [Riloff and Jones 1999] describes a two-level bootstrapping mechanism that learns item candidates and extraction rules candidates in alternating steps and verifies the candidates via a meta-level. [Collins and Singer 1999] use patterns on character level on the one hand, and syntactic contexts on the other in order to alternately train two corresponding classifiers. A study on pattern-based extraction of semantic relations from web sources as in [Duclaye et al. 1999] indicates that our method will be applicable to web data, giving rise to much larger amounts of data.

From another point of view the algorithm implements Expectation Maximisation (see [Dempster et. al 1977]) in the following way: The combination of a learning step and a verification step are iterated. When more items are found, the recall of the verification step increases.

## 4 Experiments

### 4.1 Prerequisites

Before starting the algorithm, it is necessary to set up the set of initial elements, the pattern rules and the extraction rules. For the rule part we proceeded according to sections 3.1 and 3.3 and added the rule CLASS _CAP -> NAME, with CLASS containing “Firma” and “Firmen” (“company” and “companies”, respectively). This rule is due to the observation, that relatively unknown companies are introduced in the discourse like “die Firma Demic AG, München …”

As initial set for the elements we took a list of known companies, sorted the word forms in this list by their frequency using our large reference corpus and eliminated words with either very high or very low frequency from the list. We associated most of the remaining 1,002 items with the NAME-category, 4 items of category KIND and did not use any abbreviations. Additionally, we defined the definite articles as delimiters to the left and punctuation marks and initial small words as delimiters to the right. This seed list of company name parts could be constructed from our reference corpus with reasonably little manual effort.

### 4.2 Learning by Search and Verification

Starting with the two element of CLASS and four elements of KIND (GmbH, AG, Co, KG) and with the background knowledge described in the previous section, the algorithm extracted over 12,000 items which are likely to be parts of company names. Using the extraction patterns on the sentences visited by the algorithm, about 6,000 company names could be identified.

The computational costs of an algorithm based on search and verification are rather high and can therefore only applied as an offline or background process. This is

due to the fact that a large amount of text has to be annotated and matched with the pattern rules for each item candidate: one sentence in the search step and at minimum 20 sentences in the verification step. The results presented in this paper were obtained running the algorithm on a decent PC over a weekend. Still, this does little harm to the usefulness of the results in general. Since the whole process is completely unsupervised and requires no human interactions during the run, the operational costs remain low, whereas the results can be stored elsewhere and afterwards universally applied to a wide area of applications.

### 4.3 Evaluation

The multiwords found by the extraction patterns can be divided into four categories:

1. Correct company names, like "Netscape Communications GmbH" or "Quoka Verlag GmbH & Co KG",
2. company names that are additionally prefixed by attributes like locations or specifications, e. g. "Grazer Andritz AG" or "Gabelstapelhersteller Jungheinrich AG",
3. fractions of company names like "Großmarkt GmbH" or "Datenverarbeitungssysteme mbH" and
4. errors.

Of these categories, only the latter two categories are critical. Elements of the second category may be used for extracting more specific information on the companies (for detection of additional specifiers see section 5.1). The following table gives percentage values of the four categories, obtained from manually evaluating 2'500 extracted multiterms:

| **Category** | Correct | With Specifier | Fractions | Errors |
|---|---|---|---|---|
| **Fraction** | 75.80 % | 17.36 % | 6.08 % | 0.76 % |

*Table 4: Evaluation*

While giving figures on precision is straightforward, we can merely estimate recall indirectly. Our corpus is too large to read it all through, and on substantially smaller corpora the method fails due to lack of redundancy. However, when looking at the KIND set of items, the following enumeration of legal forms of companies found in our experiment was quite satisfying:
*AG, CO, ErbStG, FilmAG, GbH, GbR, GbmH, GdbR, GesmbH, GmBH, GmbHG, GmbH, Inc, KG, KGaA, KgaA, Ltd, SpA, VVaG, aG, eG, eGmbH, eV, gGmbH, mbH, oHG*, further several composites of "*-AG*" and "*-GmbH*".

# 5 Application examples

## 5.1 Aggregation of company names

Aggregation of names requires special knowledge how authors built variants of names. First we have spelling variants (for instance, Tschibo instead of Tchibo). We have the problem to identify strings that differ by one or a few characters. Here on can use all algorithms used by spelling checkers to aggregate the corresponding items. The correct spelling should have the highest frequency.

The second problem can be described as whether to aggregate multiword candidates who differ by a complete word. If they have to be aggregated, which is the correct form?

| **Long candidate** | **Short candidate** | **Correct name** |
|---|---|---|
| Düsseldorfer Bank eG | Bank | Düsseldorfer Bank eG |
| Düsseldorfer Rheinmetall AG | Rheinmetall AG | Rheinmetall AG |
| Mannheimer Pharmexx GmbH | Pharmexx GmbH | Pharmexx GmbH |
| Infomatec Media AG | Infomatec AG | Infomatec Media AG |
| JENOPTIK Automatisierungstechnik GmbH | Jenoptik GmbH | JENOPTIK Automatisierungstechnik GmbH |
| Jenoptik Bauentwicklung GmbH | Jenoptik GmbH | Jenoptik Bauentwicklung GmbH |
| Kleindienst Datentechnik GmbH | Kleindienst GmbH | Kleindienst Datentechnik GmbH |
| Nachrichtenagentur dpa-AFX | dpa-AFX | dpa-AFX |
| Infomatec-Tochtergesellschaft Igel GmbH | Igel GmbH | Igel GmbH |

*Table 5: Aggregation of variants*

Table 5 shows long and short candidates for company names together with the correct name. There are several rules which seem to govern the examples:

Rule 1: If the first word is describes a location (for instance, a city) like in Düsseldorfer Bank eG, Düsseldorfer Rheinmetall AG, and Mannheimer Pharmexx GmbH, this location is a candidate for removal. However, the resulting shorter name should have a frequency comparable (or higher) to the long form. Because of the last condition and the non-existence of the single name candidate Bank eG, we get the correct names Düsseldorfer Bank eG, Rheinmetall AG, and Pharmexx GmbH.

Rule 2: The long candidate might contain a generic name which refers to a sector (not to a firm) like Automatisierungstechnik or Datentechnik. Usually this generic name is an interior part of the name candidate. These generic names are part of the full name. In some cases, the short candidate might refer to another firm.

Rule 3: The long candidate might contain a generic term that refers to a firm (not to a sector) like Nachrichtenagentur or Infomatec-Tochtergesellschaft. Usually this generic term is the first part of the name candidate. These generic terms describe the firm and are not part of the full name.

For all three rules, the corresponding name elements can be described using pattern or lists. Hence, they can be applied effectively.

### 5.2 Concept Aggregation in Topic Maps

One of the most challenging application areas for the proposed methodology is the use for concept aggregation within automatically generated structures from raw textual data. The general aim is to analyse an arbitrary set of documents e. g. belonging to a certain topic, institution, company or time span and extract not only prominent and relevant concepts describing the contents of these documents best, but also to specify the relationships between those concepts as a network to be exported as a XML-based Topic Map (XTM). This is accomplished by a text-mining engine that runs a complete statistical analysis of significant collocations in a given corpus (see section 1). The figure below illustrates the application on a real world example; the figure is generated in real time from the result database containing all significant collocations for the concepts in a corpus. In the example the network for "PwC" has been selected.

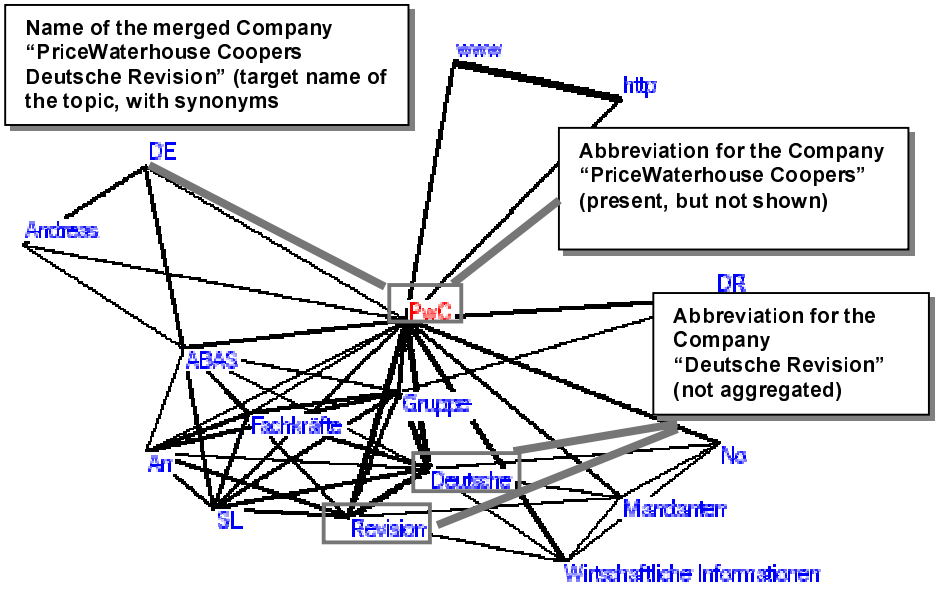


*Figure 2: Concept Aggregation in automatically created Topic Maps*

As our basic approach includes sentence segmentation into words, multiwords are not recognised in every case (although a list or already known multiwords in our database may be applied). As a result multiwords cannot be easily detected and one of the most prominent examples of interesting topics during the analysis of customer data are named entities like product or company names: The corpus above contains information about a company named "PriceWaterhouse Coopers" which is often abbreviated as "PwC". Another company that is often referred to is named "Deutsche Revision" and the corpus also contains documents of a company named "PriceWaterhouse Coopers Deutsche Revision" which is the new name of the company after the merger of the two former companies some years ago.

It is obvious that a correct analysis of concepts occurring in such a corpus of texts can only be done satisfyingly if multiword are not only recognised correctly but also identified along with their relationship among each other: For use in the TopicMap we would prefer to use an aggregated concept like “PriceWaterhouse Coopers Deutsche Revision” that comprises all the different names for a single entity in space and time (seem from the present view since we are situated in a retrieval situation). The figure above shows still the raw semantic network, which can be easily refined using the proposed algorithm to discover the compound entities.

### 5.3 Application in Named Entity Extraction (NEE)

Most of the approaches to NEE use feature vectors on word sequences rather than gazetteers [see Mladenic and Grobelnik 1998]. The argument against gazetteers is that gazetteers, no matter how big they are, will always fail on previously unseen words. Actually, this argument only holds if the gazetteer cannot be extended automatically. With a reasonable number of name elements, company name recognition can successfully work gazetteer-based in the following way: Using the name elements and the extraction rules, company names which consist of known elements only are found by pattern matching. Unseen parts of companies are detected by the pattern rules and verified in a big reference corpus, performing the verification step. Accepted candidates are added to the gazetteer.

Experiments on person name extraction were very encouraging with a precision of 97.5% at 78.2% recall, with 13’000 name elements as background knowledge. Higher recall can be obtained by simply letting the process run for a longer time.

## 6 Further Work

In the method described above, great care has to be taken in choosing highly productive and fairly precise patterns. While this task is an easy one for the case of company names, there is still the possibility of forgetting some patterns that have great impact on recall. We therefore propose an extension in the following way: the sentences used for search and verification are checked for known items that are not classified by any known pattern rule or extraction pattern. With those items new patterns are constructed and enter a test phase. Patterns that show to be productive in a way that they reappear over and over again are added to the set of patterns used for extraction and classification.

Experiments using annotated training texts are very encouraging and yield a high number of pattern rules that give rise to much larger item sets while slightly loosing in precision (see [Quasthoff et al. 2002]).

A closely related approach can be found in [Yangarber 2002], which deals with the extraction of generalized named entities in medical domains. Here, positive items do not only determine the rating of pattern rules in the test phase, but also by negative ones, which are learnt by corresponding negative pattern rules. Moreover, items and pattern rules get ratings.

## References

[Bohnet et al. 2002] Bohnet, B., Klatt, S., Wanner, L.: "A Bootstrapping Approach to Automatic Annotation of Functional Information of Adjectives with an Application to German; Proceedings of the LREC-3 Workshop: Linguistic Knowledge Aquisition and Representation - Bootstrapping Annotated Language Data, 2002, Las Palmas, Spain.

[Boehm et al. 2002] Boehm, K., Heyer, G., Quasthoff, U., Wolff, C.: "Topic Map Generation using Text Mining"; J.UCS (Journal of Universal Computer Science) 8, 6 (2002), 623-633.

[Califf and Mooney 1997] Califf, M. E.; Mooney, R. J.: "Relational Learning of Pattern-match Rules for Information Extraction"; Working Papers of the ACL-97 Workshop in NLP, 1997, 6-11.

[Collins and Singer 1999] Collins, M.; Singer, Y.: "Unsupervised Models for Named Entity Classification"; Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999, 100-110.

[Dempster et al. 1977] Dempster, A.P.; Laird, N. M.; Rubin, D.B.: "Maximum Likelihood from Incomplete Data via the EM Algorithm"; Journal of the Royal Statistical Society, Ser. B, 39 (1977), 1-38.

[Duclaye et al. 2002] Duclaye, F.; Yvon, F.; Collin, O.: "Using the Web as a Linguistic Resource for Learning Reformulations Automatically"; Proceedings of the Third Conference on Language Resources and Evaluation (LREC-3), 2002, Las Palmas, Spain, 390-396.

[Greenwood and Gaizauskas 2003] Greenwood, M. A.; Gaizauskas, R.; "Using a Named Entity Tagger to Generalise Surface Matching Text Patterns for Question Answering"; Proc. EACL 2003 Workshop on Natural Language Processing for Question Answering, http://remote.science.uva.nl/~mdr/NLP4QA/09greenwood-gaizauskas.pdf.

[Heyer et al. 2001] Heyer, G.; Läuter, M.; Quasthoff, U.; Wittig, Th.; Wolff, Ch.; "Learning Relations using Collocations"; Proc. IJCAI Workshop on Ontology Learning, Seattle/WA, August 2001, 19-24.

[Mladenic and Grobelnik 1998] Mladenic, D.; Grobelnik, M.: "Feature Selection for Classification Based on Text Hierarchy"; Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98, 1998.

[Nahm and Mooney 2002] Nahm, U. Y.; Mooney, R. J.: "Text Mining with Information Extraction"; Proceedings of the AAAI 2002 Spring Symposium on Mining Answers from Texts and Knowledge Bases, Stanford/CA, March 2002, 60-67.

[Quasthoff et al. 2002] Quasthoff, U.; Biemann C.; Wolff, C.: "Named Entity Learning and Verification: Expectation Maximisation in Large Corpora"; Proc. 6th Conf. On Natural Language Learning 2002 (CoNLL-2002), Taipeh, August 2002, 8-14.

[Quasthoff and Wolff 2000] Quasthoff, U.; Wolff, Ch.: "An Infrastructure for Corpus-Based Monolingual Dictionaries"; Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation, Athens, May / June 2000, Vol. I, 241-246.

[Riloff and Jones 1999] Riloff, E.; Jones, R. (1999): Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. Proceedings of the sixteenth National Conference on Artificial Intelligence (AAAI-99), 1999, 1044-1049.

[Yangarber et al. 2002] Yangarber, R.; Lin, W.; Grishman, R.: "Unsupervised Learning of Generalized Names"; Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002), Taipei, Taiwan.