A Lexically–Intensive Algorithm for Domain–Specific Knowlegde Acquisition

René Schneider *
Text Understanding Systems
Daimler-Benz Research and Technology
Ulm, Germany
rene.schneider@dbag.ulm.DaimlerBenz.COM

Abstract

This paper is an outline of a statistical learning algorithm for information extraction systems. It is based on a lexically intensive analysis of a small number of texts that belong to one domain and provides a robust lemmatisation of the word forms and the collection of the most important syntagmatic dependencies in weighted regular expressions. The lexical and syntactical knowledge is collected in a very compact knowledge base that enables the analysis of correct and partly incorrect texts or messages, which due to transmission errors, spelling or grammatical mistakes otherwise would have been rejected by conventional systems.

1 Introduction

The major tasks of information extraction systems (IE–Systems) are the unsupervised selection, fast analysis and efficient storage of relevant text patterns a person or a group of persons is interested in. It accomplishes this through the use of learned or handcrafted patterns. In the ideal case the results lead to an appropriate reaction, executed by the computer itself (see Figure 1). The extracted information is stored in a template that usually is based on a slot–and–filler model. Whenever the textual information does not fit automatically into the fillers, it has to be changed adequately to the form and content requirements of the template, otherwise the text is rejected. Thus, the templates architecture depends very much on the domain the IE–system was built for, i.e. before processing a text or a message and starting the linguistic analysis, the category that the text or message belong to is already known or has been labeled automatically with the aid of a categorizer. In our investigation the system was built to process requests for business reports, extracting the number, years and language of business reports a certain sender asked for.

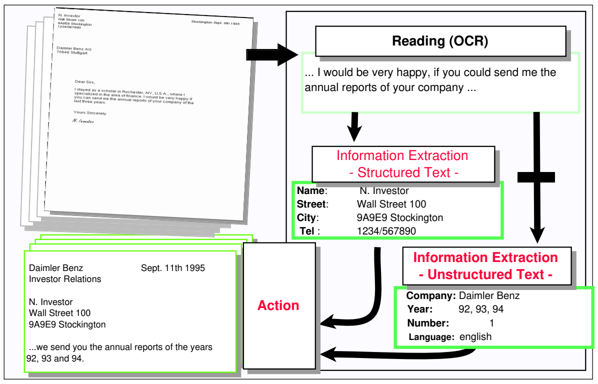

Figure 1: Overview of an Information Extraction System

Although a lot of sophisticate investigation has been done in the area of information extraction (Pazienza, 1997) (esp. since the start of the MUC–Conferences in 1987), only few works are concerned with the automatic acquisition of the knowledge bases that are needed for IE–tasks (Riloff, 1993), which makes the construction of a new system for a different extraction task still very expensive and says much about the brittleness of "traditional" IE–systems . The problem gets worse when the information that has to be extracted is paperbound and has to be digitized by scanners to make the information available to the computer, because Optical Character Recognition (= OCR) still garbles a considerable amount of information reduction and noise on texts, so that there is also a need for more robust information extraction systems that handle

* This study is part of the project READ. The project READ is funded by the German Ministry for Education and Research (BMBF) under grant 01IN503C. The author is responsible for the contents of the publication.

René Schneider (1998) **A Lexically-Intensive Algorithm for Domain-Specific Knowledge Acquisition.** In D.M.W. Powers (ed.) *NeMLaP3/CoNLL98: New Methods in Language Processing and Computational Natural Language Learning,* ACL, pp 19-28.

noisy information adequately.

The work presented in this paper reflects a statistical approach for the automatic acquisition of a linguistic knowledge base, that allows the essential analysis for texts of a certain domain, independent of their transmission quality or pre–processing.

2 Challenges in Information Extraction

2.1 The Acquisition Bottleneck

Generally, IE–Systems are built for a rather restricted task and work on a more or less limited domain. This keeps their knowledge bases and the rules that are needed to process the texts, e.g. the syntactic rules, quite compact. But nevertheless, the changes that have to be done whenever a working system is applied to another domain are remarkably high, in some cases leading to the construction of a almost completely new knowledge base.

Both, the construction of a new knowledge base and their maintenance need a certain time and lots of efforts have to be done by highly–skilled staff knowing the system and the domain it is built for. On the other hand, texts or messages that are written for a very specific purpose show the phenomena of Sublanguages (Harris, 1982), with less ambiguities and varieties than unrestricted language but still more freedom in expression than Controlled Languages. This fact strengthens the need for the automatic acquisition of linguistic knowledge, esp. the construction of a lemmatisation and a shallow parsing component.

Statistical learning algorithms are usually applied to processing large corpora, but in real life, huge samples are hard to find for commercial and industrial applications. In our case, the corpora usually consist of small samples of fewer than 150 very short texts and the whole sample must be split into a training and a test corpora. This disadvantages are compensated by the use of a domain-specific sublanguage. Any sublanguage shows some use of typical vocabulary, styles, and grammatical constructions, and it can be said that the more specific the domain is, the stronger are the restrictions of the sublanguage. But even in categories where these restrictions are weak, the essential and relevant information is carried by some typical words and located in a few kernel phrases, so that even simple statistics like frequency lists, distance measures and weighted collocation patterns may overcome parts of the acquisition problem (see section 4).

2.2 The Noisy–Channel–Problem

The second major problem is concerned with the fact that still a remarkably high number of paper-bound texts have to be pre–processed by an OCR–System in order to convert them into machine-readable code. This problem can be compared to the well known problem of a noisy channel, as indicated in Figure 2.

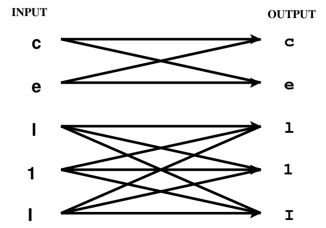

Figure 2: An Example of a Noisy Channel in OCR

Therefore, the development of OCR systems and the improvement of their efficiency is still a major task in the area of document processing. But even with high quality scanners, the promised 99.9% recognition rate is difficult to achieve (Taghva et al., 1994) and remains the ideal case due to e.g. the use of different fonts, low quality print or paper, a low resolution etc.

Besides the mistakes caused by OCR, a considerable number of documents include typographical or grammatical mistakes (misspellings, wrong inflection or word order), unusal expressions etc., which shows that natural language processing (NLP) needs more than just a grammar for grammatical expressions but indeed has to be fault–tolerant to process "real–world " utterances. Though natural language itself has a lot to do with exceptions and irregularities, all these nuisances amplify the problems NLP is occupied with, but — as a glance at text samples shows — IE–systems are faced with a considerable number of these additional irregularities that occur

- as a result of low grammatical competence, e.g. whenever a non–native speaker is obliged to write a document or message in a second language;
- as careless slips, e.g. misspellings, missing punctuations etc.

However, the most of all occurring errors are produced by OCR[1] and can be classified as follows:

[1] A brief example of an OCR-text: `We would be vBry pleasd ifyou could send two 1992 annuai reports and a product brochure to: ···`

- Incorrect character recognition:
  - Merging or Splitting: Two or more characters are represented as one and vice versa.
  - Replacement: Characters are confused, e.g. 1 and l.
  - Deletion: Characters are dropped (e.g. due to low print quality).
  - Insertion: Non-existing characters are added.
- Incorrect word boundary recognition:
  - Agglutination: Two or more word boundaries are not recognized, and distinct words are linked with each other.
  - Separation: A single word is split into two or more fragments.

Therefore, in this study one of the principle goals was to find a new methodology that enables the computer to learn automatically from a very small data set with examples of both grammatically incorrect and orthographically ill–formed text.

3 Machine Learning in Information Extraction

3.1 Statistical Language Learning

Machine learning techniques have been developed to acquire factual and conceptual knowledge automatically and all of them have been applied to natural language processing. The different techniques were derived from the fields of symbolic, connectionist, statistical and evolutionary computing and their application depends on the specific problem. Recent developments show that the consecutive or simultaneous combination of different learning approaches, i.e. hybrid strategies, often leads to better results than the single use of one. The methodology most frequently used to support other learning strategies are statistics, but in several occasion they are also used exclusively, esp. when the *a priori* knowledge about the content and the structure of the data is very low (Vapnik, 1995).

In such cases, all that is needed to start with, is the knowledge about some functional properties of the data to deduce their dependencies. Simply speaking, an unordered or hidden structure is transformed into a systematic structure revealing the properties, relations and processes of the data. In the ideal case the discovery of these dependencies leads to the formulation of general principles or laws.

In NLP, statistics are used to describe the processe of language acquisition, language change or variation (Abney, 1996) using the methods of information– and probability theory (Charniak, 1993). Thus, the starting point of every investigation discovering these processes in order to "learn" a language or acquire knowledge about some language with statistical techniques is the likelihood of words and their derivable distributions and functions.

MIROMIR, an independant financial and economic research society, is making a study about Leasing in Europe. In order to make a prvsentation of your company, *we* would like to **recieve** your commorcial documents and your *last* **annual roports** (*from 1988 to 1991*) in *english*. If you have a **mailing list** would you kindly **include** *our name* for future issues of **annual repords** and lnformation on your company. With our grateful thanks, yours faithfully.

Figure 3: **Domain–Specific** and *Text–Relevant* Information (OCR–text)

3.2 Domain-Specific and Text-Relevant Knowledge

Besides that, the formulation of *what* has to be learned needs to be formulated and described precisely, esp. in IE where the different elements of the whole data set are not regarded with the same degree–of–interest and only a very small part of the whole information is extracted. Hence, the system has to learn to divide between the important or interesting and the unimportant or less interesting information. In case of OCR–errors, it has to be able to clean the text from noisy parts and restore those parts appropriately.

The interesting parts of a text or a message, which have a high significance for IE–systems, can be divided into domain–specific and text–relevant data (or high level and low level patterns (Yangarber and Grishman, 1997)) as illustrated in Figure 3, where the domain–specific words are represented in bold and the corresponding text–relevant information in cursive letters. The domain–specific words can be seen as distinctive from all other words since they describe the domain and general purpose the text has been written for, whereas the text–relevant words stand in a close relation to the domain–specific data because they usually do not appear alone but determine exactly the meaning of the domain–specific words. In the case of our example in Figure 3 the domain–specific information is represented by the words `recieve, annual roports` and `include, mailing list`. The text–relevant information `MIROMIR, we, from 1988 to 1991, english, our name` specifies the numbers, years and language of the annual reports requested and of course the sender (which in the case of `we` and `our name` has to be unriddled by anapher resolution) that should be included into the mailing list.

To illustrate the relationship between domain–specific and text–relevant information, their functions may be compared to those of constants and variables in a mathematical equation with the domain–specific words (representing the unvariable and basic components of the equation) and the text-relevant information representing the variables (as unstable and characteristic elements of the equation). Thinking in categories of natural language, the domain–specific information represents the pragmatic meaning and uses verbs and specific nouns to describe specific events while the text–relevant information is represented through names, numbers, dates etc. In any case it has to be considered that this distinction depends very much on the sharpness of the domain the IE–System is built for. Generally speaking, the more specific a domain is, the better does this distinction work and thus facilitates both the construction of the output structure (templates) and the extraction of the relevant text features.

Unfortunately — as will be seen in the next section — text-relevant information is very difficult to learn automatically, particularly when the texts that are analyzed have been damaged by OCR: e.g. the differences of `report` and `roport` can easily be detected and resolved, whereas names of persons, streets etc. themselves have several spelling variants and the change of a single letter changes the whole meaning, as it happens for numbers, too.

Therefore the main focus is on the the detection of domain–specific information with statistical methods that leads in a following step to the text–relevant information, i.e. the major task of the algorithm is to build automatically a knowledge base for the crucial words and the kernel phrases that represent the salient information of a given text.

4 Lexically–Intensive Knowledge Acquisition

4.1 An Outline of the Algorithm

The algorithm as illustrated in Figure 4 is divided into the following major steps: First, a frequency list is computed from the training data, i.e. the raw text of a limited number of texts belonging to the same domain. Then, all word forms are compared with each other and the word forms with a low distance are grouped together. The results from these two procedures are combined and lead to the construction of a very compact core lexicon that consists of a limited number of entries with lexical prototypes and automatically assigned variants of the corpus' word forms. Afterwards the training data is trans-

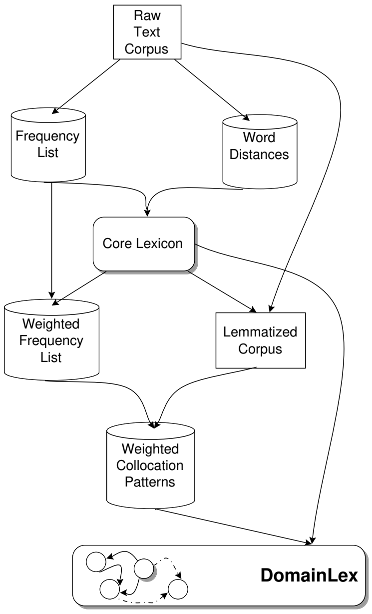

Figure 4: Building a Domain-Specific Lexicon

formed so that it only consists of the automatically derived lexical prototypes. Then the most frequent syntagmatic patterns from a length of two to five lemmata are collected and weighted. In the last but one step similar patterns having at least one domain-specific lexeme in common are collected to reveal the neighbourhoods of the most important words. The degree–of–interest of a word is computed from its frequency and the number of variants the word has. Finally the entries of the core lexicon are connected with one another and compressed into weighted regular expressions. The result is a domain–specific lexicon that is represented as a net of lexical entries covering the correct word and their variants and some of the possible incorrect variants and the syntactical relations that are commonly used in texts of a certain domain.

4.2 Acquisition of Lexical Knowledge

The construction of the core lexicon is based on the combination of a frequency list and a comparision of the distances of all word forms given in a corpus of ca. one hundred texts. [2]

A computation of the relative size of unknown and already known word (see Figure 5) shows that after a very low number of texts generally 80 % of the information is confirmed, i.e. it appeared already in one of the former texts. These 80 % cover generally the functional words such as articles, conjunctions etc. and of course the domain–specific information. The residual 20 % consist of text–relevant information, unimportant and less interesting information, misspellings, and — in OCR–texts — noisy information.

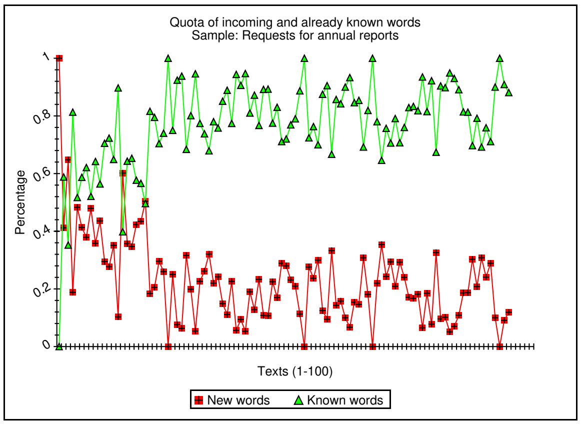

Figure 5: Incoming and Known Words in a Sequentially Ordered Training Corpus

A closer look at the frequency lists strengthens this impression and allows the postulation of the following hypothesis:

Hypothesis 1 *The more frequent a word appears in a number of consecutively ordered texts or messages of a limited domain, the more probable will it represent the "lexical prototype" for a wordform and in OCR–texts the correct form of a prototype (or a lemma).*

To find out which possible variants exist for the whole number of word forms, the similarities (or distances) of the word forms are computed. An effective method for the measurement of word distances is the Levenshtein distance in combination with an adequate threshold value (Nerbonne et al., 1996),

Word	Variants	Distance	Frequency
rbport			1
	report	0.333	89
	reports	0.428	62
	roports	0.428	1
	ofreports	0.555	1
	reporting	0.555	2
	reporting•	0.6	1
	•xport	0.666	1
	sports	0.666	1
	cort	0.666	1
	fjeport	0.714	1
	portfolio	0.777	1
	important	0.777	1
	importance	0.8	1
	portfolios	0.8	1
	opportunity	0.818	3
	north	0.833	1
	opportunities	0.846	2

Table 1: Unordered Lexicon Entry: `rbport`

since the operations that are done to calculate this distance cover most of the phenomena (see 2.2) that occur through OCR. Any two words are compared with each other in a distance matrix, which measures the least effort of transforming one word into the other. Least effort means the lowest number of insertions, deletions, or replacements (as a combination of deletion and insertion). The effort is normalized to the length of the longest word in order to obtain a ratio-scaled value. Table 1 gives the example of an unordered lexicon entry for the word form *rbport* with all similar words that were found in the corpus, having a Levenshtein distance lower than 0.9 [3]. As already postulated in Hypothesis 1, the number of correct and "deflected" forms is always higher than those of *typical* OCR-mistakes. In fact it must be asked, whether typical OCR mistakes exist at all due to the different types of reasons for these mistakes and the multitude of effects they may have.

For every word with one or more similar words as determined by a threshold value of 0.9, a preliminary entry was created as illustrated in Table 1, covering the most important morphological deriva-

[2] In this paper we focus on the results of a training corpus of business letter requests in english with a total number of only 7,078 word forms distributed over 100 texts. Notice that the average size cannot be regarded as statistical significant due to the standard deviation of 42.32 and the text sizes ranging between 15 and 256 tokens.

[3] To facilitate and shorten the work of the algorithm, the alphabet was divided into interpretable *signs* (a–Z , 0–9, punctuation) and non-interpretable *signs* (like * , ˜ , ˆ etc.) which were converted into a middle point (•) . A *word* is considered to be everything between two empty spaces. A *text* or text body is everything that remained on the document after the elimination of head and foot structures (e.g. sender, address, signature, etc.)

tions and graphemic alternations, whereas in none of the entries a distinction between lemma and variants is made so that the unordered lexicon bears a huge burden of redundant information. To diminish this redundancy, it is necessary

- to drop those words having a high distance and showing no linguistic relation to the other words in the entries and
- to make a clear distinction between a lemma and its variants.

Therefore the multitude of preliminary lexical entries was reduced to a very compact core lexicon as exampled in Table 2 and described as follows.

The algorithm processes successively through the frequency list, starting with the most frequent word and finishing with the last hapax legomenon. Each word that can be found in the frequency list is considered as the top of a new lexicon entry or lemma. Afterwards, the algorithm looks for the word forms in the preliminary lexicon, that are similar to this word (having a distance smaller than 0.4), assigns them as variants in the new entry and recursively looks for all variants of the previously assigned variants (having a distance smaller than 0.7). Each one of these variants can no longer be regarded as top of another entry and consequently is taken out of the frequency lists, that simultaneously shrinks more and more. The variants' frequency is added to that of the lemma.

The results of the algorithm depend a lot on an *a priori* specified treshold value for the Levenshtein distances. In our tests, good results are achieved with a value of 0.4 for direct similarity and 0.7 for indirect similarity, meaning the newly computed distance of variants of a variant to a given lemma. The threshold value may depend on the language and the domains that are used. This aspect will be further investigated.

The result of this process is a core lexicon that consists of

- high frequent synsemantica or function words having no variants,
- high frequent, domain specific autosemantica or content words and most of their occuring variants,
- middle and low frequency words and their variants, and
- one single entry for all the remaining hapax legomena having no similarity to one of the preceding words lower than 0.4,

Stem	Variants	Distance	*f req*
report			88
	reports	0.142	61
	reprt	0.166	2
	repo	0.333	1
	rep0rt	0.333	1
	rbport	0.333	1
	ofreports	0.333	1
	reporting	0.333	2
	reporting•	0.4	1
	roports	0.428	1
	fjeport	0.428	1
	repoii	0.666	1
	sports	0.666	1
	repods	0.666	1
Σ	13		163

Table 2: Core Lexicon Entry: *report*

in order of their summarized frequencies. Hence, the number of entries in the core lexicon is at about one third of the total number of types[4]. Table 2 shows the entry for `report` and the assigned variants.

As follows, many of the wrongly analyzed combinations of e.g. *your annual report* that formerly lead to a rejection of the text, now can be transformed into their correct forms. This increases the number of documents that can be analyzed by the IE–system considerably. The wrong assignment of `sports` as a variant of `report` shows the domain dependency of the algorithm, but it has to be considered that the frequency of such wrong assignments generally is 1 and can be compensated by the extraction of syntactical patterns.

4.3 Acquisition of Syntactical Knowledge

The core lexicon bears the basic lexical knowledge that is needed for a morphological text analysis and furthermore can be used to "clean" documents from noisy sequences but it does not store any information about the syntagmatic relations or dependencies that exist in texts of a given domain. To reveal these dependencies, the original corpus was transformed into a lemmatized version, consisting only of the earlier derived prototypes and "Weighted Ranks" for words with the frequency 1 having no similarity to other words. Figure 6 shows the example text (see Figure 3) after the transformation into lemmata and "jokers". As can be seen in Figure 6, the algorithm

[4]In the case of the english requests for annual reports the core lexicon comprised 537 entries with a total number of 1758 types and 7078 tokens in the training corpus.

```
- , an independant financial and
economic research - , is making a study
about leasing in european .  in order to
make a - of your company , we would like
to receive your commorcial documents
and your latest annual report from - to
1991 in english .  if you have a mailing
list would you kind include our name
for future issues of annual report and
information on your company .  with our
grateful thank , your - .
```

Figure 6: A Lemmatized Text

- suppresses (in this format) the Hapax–Legomena like `Miromir`, `society`, `prvsentation`, 1988 and `faithfully`;
- corrects the OCR-errors of the most important words, like `roports` → `report`, `repods` → `report`, `lnformation` → `information`;
- corrects misspelled words, like `recieve` → `receive`;
- lemmatizes several less frequent words to their more frequent prototypes, like `ropods` and `repods` as plural forms of `report`, `last` → `latest`, `kindly` → `kind`, `thanks` → `thank`, `yours` → `your`[5].

To enhance the importance of the lexical protoypes or lemmata, their frequencies were multiplicated with the number of their assigned variants as a result of the following hypothesis:

Hypothesis 2 *The more often a word appears in texts of a restricted category and the more morphological and graphemic variants it has, the more probable the word will represent some domain-specific information.*

The multiplication of frequencies and the number of variants of a word ($freq_x \quad var_x$) leads to a weighted frequency list (see Table 3) whose first ranks comprise the most relevant lemmata that are needed for the extraction of the salient syntactic patterns. Therefore, the texts are transformed parallely into a corpus of indices implying the ranks that are given to the lemmata after they have been weighted.

[5] `commorcial` is head of an entry including `commercial` as the single variant, both having the frequency 1. Thus, the distinction between stem and variant can not be done clearly by the algorithm (the same holds true for `independant` and `independent`). Nevertheless, the two forms and all newly occuring forms having a small distance value will be clustered together.

Rank	Word	$freq_x$	var_x
1	report	163	13
2	annual	117	10
3	your	208	5
4	would	175	5
5	thank	58	8
6	company	70	5
7	other	40	8
8	mailing	51	6
9	information	44	6
	please	66	4
10	financial	36	7
11	grateful	33	7
12	the	209	1
	to	209	1
13	statements	26	8
14	international	29	7
15	of	201	1
16	latest	45	4
17	list	42	4
18	you	166	1
19	and	164	1
20	receive	27	6

Table 3: Weighted Frequency List (first 20 Ranks)

The concluding analysis follows the Firthian notion of "knowing a word by the company it keeps" (Firth, 1957), a postulate which emphasizes the fact that certain words have a strong tendency to be used together. Thus, the algorithm retrieves all collocation patterns of different length (2 - 5) and matches them with one another. Repetitively the most frequent patterns are matched with the collocation patterns of a greater length (patterns of length 2 with patterns of length 3; patterns of length 3 with patterns of length 4 etc.) looking both left and right for high frequent lemmata in the neighbourhood of the already composed patterns. That means that the words from the top of the weighted frequency list are connected with the most common words that precede and succeed them. The result is a two–way finite-state automaton that may be analyzed using light parsing strategies (Grefenstette, 1996) with the salient words of the weighted frequency lists as starting points (see Figure 7).

One attractive alternative to parse the text is a bottom–up island parser for the kernel phrases of a new text. Island Parsers are a useful tool especially in those cases where no sentence markers exist (as e.g. in speech recognition) or whenever they are not transmissed correctly or added (as in OCR-texts). Furthermore a full parse contradicts in a certain way the real ambitions of IE–Systems (Grishman, 1996) and flat finite-state analyses are getting more and more popular and efficient (Bayer et al., 1997) . Yet,

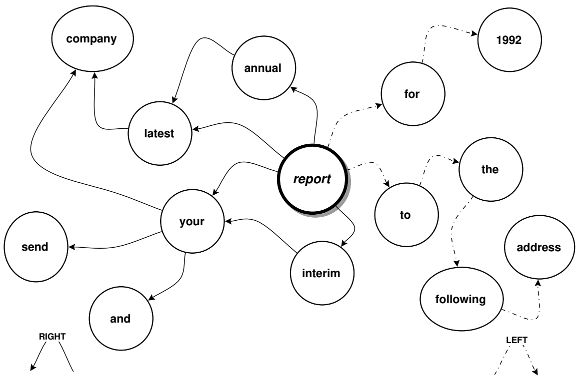

Figure 7: Generation of a bi–directional regular expression

the statistical information that is represented in the ranks of the lexical stems should not be omitted, though they show evidence of the degree–of–interest that is needed for the parsing strategy. The neighbourhood of low indices, such as $3 \leftarrow 2 \leftarrow 1$ (representing `your annual report` should be regarded as more representative for the corpus' syntax than e.g. $15 \leftarrow 3 \rightarrow 6$ representing `of your company` or even $95 \leftarrow 45 \rightarrow 381$ representing `am currently doing`. The weighted ranks represent the degree–of–interest that the words have for the IE–System. With the help of the weighted ranks, it is possible to compute a probabilistic value similar to transition likelihoods. Looking at a pattern or window of several words w_i of a given pattern length n, we add up the ranks of the weighted frequency lists $\tilde{r}_{w_i}$ to $\tilde{r}_{w_n}$ and compute the average rank. This value is divided by the overall frequency $freq$ of the whole pattern $(w_1..w_n)$:

$$\tilde{C}_{(w_1..w_n)} = \frac{\frac{\sum_{i=1}^{n} \tilde{r}_{w_i}}{n}}{freq(w_1..w_n)} = \frac{\sum_{i=1}^{n} \tilde{r}_{w_i}}{n\ freq(w_1..w_n)}$$

The resulting value represents the weighted likelihood for the co–occurrence $\tilde{C}_{(w_1..w_n)}$ of two (or more) words indicating how probable a word precedes or succeeds another word. To give an example the word pattern of two words like `mailing list` the equation is solved a follows:

$$\tilde{C}_{(mailing\ list)} = \frac{\tilde{r}_{mailing} + \tilde{r}_{list}}{2\ freq(mailing\ list)} = \frac{8+19}{2 \cdot 35} = 0.385$$

or for longer patterns of a lower degree–of–interest, such as `as any interim`:

$$\tilde{C}_{(as\ any\ interim)} = \frac{\tilde{r}_{as} + \tilde{r}_{any} + \tilde{r}_{interim}}{3\ freq(as\ any\ interim)} = \frac{53+87+28}{3 \cdot 1} = 56.0$$

As already pointed out, the values for the co–occurrences of the different lemmata were only computed up to a length of 5 lemmata. Compared to other collocation measures, this value does not only take account of the words frequencies and the collocations frequencies (as e.g. Mutual Information (Church and P., 1990)) or their transition likelihood (as e.g. Markov chains (Thomason, 1986)) but combines these two properties with a third one: the word's different modalities as indicated by their number of variants, i.e. their weighted ranks. This last value weakens the influence of both less frequent and functional words and supports the degree–of–interest of domain-specific and correct words as determined in Hypothesis 1 and 2.

The co–occurrence values may be labeled to the arcs of the regular expressions that are generated during this acquisition process to make the parsing process more effective since a low transition value reflects a high significance or degree–of–interest in texts of a certain domain.

5 Using the Domain-Specific Lexicon

The connections that exist between the different lexical entries are also used to link the entries of the core lexicon, providing it with the syntactical information that is typical for a certain domain. The contents of the entries and their relations, i.e. the arcs connecting them, cover the essential statistical properties of lexemes and their syntactical relationship, enabling a robust lexical and syntactical analysis of new texts.

First results show that word forms are deflected

```
··· in the past your companys report has
been among those we collect .  however
, our records indicate we do not have
a copy of your 1992 annual rbport .
please help us complete our collection
by sendhig a copy of your 1992 annual
report to the followhig adess .  ···
```

Figure 8: Example of an Unknown Text (OCR)

and corrected (as shown in Figure 6 and 9); kernel phrases are isolated by extracting the islands of the domain–specific words and their surroundings (as

```
··· in the - your company report has been
among these we collection .  however ,
our records indicating we do not have
a copy of your 1992 annual report .
please help us complete our collection
by sending a copy of your 1992 annual
report to the following address ···
```

Figure 9: Lemmatizing an Unknown Text

shown in Figure 7 and 10). Although some words are lemmatized in a quite strange way, as e.g. `collect` → `collection` or `indicate` → `indicating` due to their low frequence, the relevant patterns are converted to analyzable and well formed strings.

Given a new text with several occurences of a highly–ranked words, (see Figure 8), the text is lemmatized (see Figure 9 and browsed for the word with the highest degree–of–interest as indicated by the words' weighted ranks (in our example `report`).

Afterwards the transition values for the three neighbourhoods of `report` are compared and ordered after the values of the weighted transition likelihood (see Figure 10 with immediate transition values i.e. a window length 2). In our case, the second phrase has the lowest transition values and would consequently extract and parse succesfully the most relevant phrase `a copy of your 1992 annual report to the following address`.

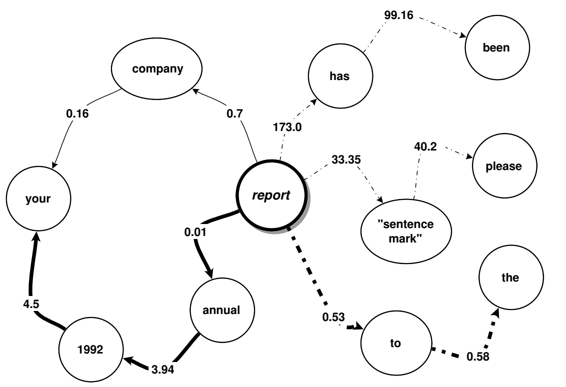

Figure 10: Parsing the Most Likely Neighbourhood

The lexicon's dynamic structure enables the analysis of unknown texts and consequently updates the entries and the relations among them, i.e. whenever an unknown word or a new syntactical pattern appears, the Levenshtein-Distance to the already existing heads of the lexical entries is computed and the word either stored as a new variant or a new entry created. Similar to the lexical updating process the weights of the tokens that connect the lexical entries are either affirmed and strengthened with the repetition of every pattern that was already known to the system or the new pattern is added to the network.

6 Conclusion

In this paper we discussed the construction of a statistical learning algorithm based on restricted domains and their underlying sublanguages in order to build automatically a linguistic knowledge base for information extraction tasks with the aid of very simple arithmetic procedures. The method is based on weighted frequency lists of word forms and syntactical patterns. Although very small information about the texts and the domain is known *a priori* and only two functional dependencies (see Hypotheses 1 and 2) have been postulated, the algorithm learns automatically to build a very compact knowledge base from small and noisy text corpora. The method was tested empirically on several english, german and spanish corpora and shows the same results for noisy as well as for correct domain–specific corpora.

A comparison of the core lexicon with common frequency analyses (Francis and Kučera, 1982) for correct texts shows that even with a very small text sample the resulting information for linguistically allowed alterations of a lexical base form is acquired automatically. Additional information is achieved with the subsumption of linguistically incorrect variants. The acquired knowledge is stored in a compact and dynamic knowledge base whose structure is modified with every significant change of the lexeme's probabilistic properties and relations. The knowledge base is quite compact and allows a very quick analysis of unknown texts.

First tests with different corpora and different languages (German and Spanish) show that this algorithm can be applied to different domains and other languages and thus is a useful tool for the expansion of IE–systems that work with OCR–data. Although the results of the algorithm depend very much on the data, i.e. the limits or sharpness of the domain which is used, the underlying ideas may be used for any information extraction purpose and other applications such as lexicography, information retrieval or terminology extraction.

7 Acknowledgements

I wish to thank Ingrid Renz and Uli Bohnacker for all the ideas, suggestions and comments that found their way into this paper.

References

Steven Abney. 1996. Statistical methods and linguistics. In Klavans, J.L. and Resnik, P., editors, *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, pages 1–26. MIT Press, Cambridge, MA.

Thomas Bayer, Uli Bohnacker and Ingrid Renz. 1997. Information extraction from paper documents. In Bunke, H. and Wang, P.S.P., editors, *Handbook on Optical Character Recognition and Document Image Analysis*, pages 653–677. World Scientific Publishing Company, Singapore.

Eugene Charniak. 1993. *Statistical Language Learning*. MIT Press, Cambridge, MA.

Kenneth W. Church and Patrick Hanks. 1990. Word association norms, mutual information and lexicography. *Computational Linguistics*, 16(3):22–29.

John R. Firth. 1957. Modes of meaning. In *J.R. Firth: Papers in Linguistics*, pages 190–215, London. Oxford University Press.

W. Nelson Francis and Henry Kučera. 1982. *Frequency Analysis of English Usage*. Houghton Mifflin, Boston, MA.

Gregory Grefenstette. 1996. Light parsing as finite-state filtering. In *Proceedings of the Workshop on Extended Finite State Models of Language, ECAI'96*, Budapest, Hungary.

Ralph Grishman. 1996. The NYU system for MUC-6 or where's the syntax? In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, Columbia, MD. Morgan Kaufmann.

Zellig S. Harris. 1982. Discourse and sublanguage. In Kittredge, R. and Lehrberger, J., editors, *Sublanguage: Studies of Language in Restricted Semantic Domains*, pages 231–236. de Gruyter, Berlin.

John Nerbonne, Wilbert Heeringa, Erik van den Hout, Peter van der Kooi, Simone Otten and Willem van de Vis. 1996. Phonetic distance between dutch dialects. In Durieux, G., Daelemans, W., and Gillis, S., editors, *Proceedings of Computational Linguistics in the Netherlands*, pages 185–202, Antwerp, Centre for Dutch Language and Speech (UIA).

Maria Teresa Pazienza. 1997. *Information Extraction - A Multidisciplinary Approach to an Emerging Information Technology*. Springer, Berlin.

Ellen Riloff. 1993. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the 11th National Conference on Artificial Intelligence, AAAI-93*, pages 811–816.

Kazem Taghva, Julie Borsack, Allen Condit and Srinivas Erva. 1994. The effects of noisy data on text retrieval. *Journal of the American Society for Information Science*, 45(1):50–58.

Michael G. Thomason. 1986. Syntactic pattern recognition: Stochastic languages. In Fu, K.S. and Young, T.Y., editors, *Handbook of Pattern Recognition and Image Processing*, pages 119–142. Academic Press, Orlando, FL.

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer, New York.

Roman Yangarber and Ralph Grishman. 1997. Customization of information extraction system. In *Proceedings of the International Workshop on Lexically Driven Information Extraction*. Universitá di Roma "La Sapienza".