Information Extraction by Text Classification: Corpus Mining for Features

Jakub Zavrel*, Peter Berck†, Willem Lavrijssen†

*CNTS / Language Technology Group
University of Antwerp
Universiteitsplein 1
2610 Wilrijk, Belgium
zavrel@uia.ua.ac.be

†Stichting Toepassing Inductieve Leertechnieken (STIL)
Eisingahof 37
5025 DN Tilburg, The Netherlands

Abstract

This paper describes a method for building an Information Extraction (IE) system using standard text classification machine learning techniques, and datamining for complex features on a large corpus of example texts that are only superficially annotated. We have successfully used this method to build an IE system (Textractor) for job advertisements.

1. Introduction

For rapid development of an Information Extraction system in a large new domain, the usual methods of semi-corpusbased hand-crafting of extraction rules are often simply too laborious. Therefore one must turn to the use of machine learning techniques and try to induce the knowledge needed for extraction from annotated training samples. Techniques for the induction of extraction rules are e.g. described by (Freitag, 1998; Califf and Mooney, 1999; Soderland, 1999). To learn extraction rules from examples, the training samples must normally be annotated in a detailed manner, so that each entity to be extracted is marked in its exact location in the text. Rules can then be induced by generalizing contexts of occurrence of the relevant entities.

Unfortunately, it is a lot of work to manually annotate a large enough sample of texts at this level of detail. However, organizations interested in automating *existing* Information Extraction tasks often do have large quantities of texts paired with a database of category-labels for relevant entities (see Figure 1). This allows us to rephrase the Information Extraction task (extract slot-filling strings out of text) as a set of Text Classification tasks (assign category labels to the text; one for each slot).

In the setting that we worked in, about 50 thousand job advertisements, categorized by job title, education requirement, and industry sector, were available (each coded by a large number of fine-grained categories). Thus the three types of extraction slots were replaced by three separate classification tasks.

For text classification, we can train existing machine learning techniques to assign category labels on the basis of features contained in the texts. This means that we treat a symbolic representation of the entities to be extracted as categories, and the actual instantiations of these entities and strongly predictive co-occurring strings as features.

Since this rephrasing of the IE task results in a large amount of category labels, and an even larger amount of instantiations and cues for those labels, we are forced to pay special attention to an important issue in text classification, viz. the representation of documents in terms of features. Usually, one starts with a bag of all the terms in a document, and selects some subset of these that seem to be good predictors for the categories (see e.g. (Yang and Pedersen, 1997; Mladenic, 1998)). This is also our starting point, but we gradually extend the document representations with more complex features, based on a semi-automatic search in the training data for good predictors.

The search for good features is not restricted to single words from the documents. In previous work, good results have been obtained using longer phrases as features (Mladenic and Grobelnik, 1998; Spitters, 1999). For example, whereas “network” and “engineer” are very generic terms in a corpus of job advertisements, “network engineer” points to a very specific category.

The data that we had available contained an additional source of information: labeled section boundaries. Hence we were also able to consider more complex features, viz. conjunctions of a section label and a phrase. This rather rich feature language gives us possibilities to achieve high precision. The price we pay for this is that we must face a very large number of potential features. This lead us to a treatment of feature selection as a data-mining task in the example corpus. We have experimented with a number of existing and new feature-selection methods, some based on simple corpus statistics, and some on a human-assisted discovery process in the corpus.

Using this methodology, we have been able to build Textractor, a fully functioning, and reasonably accurate, IE system for Dutch job advertisements, in about four months.

In this paper we describe the basic philosophy behind the system, and report the results of some experiments. The remainder of the paper is structured as follows. Section 2. gives a description of the data set and the classification tasks that were defined on this data. Section 3. briefly presents Memory-Based Learning, the machine learning algorithm used for classification. Section 4. describes the

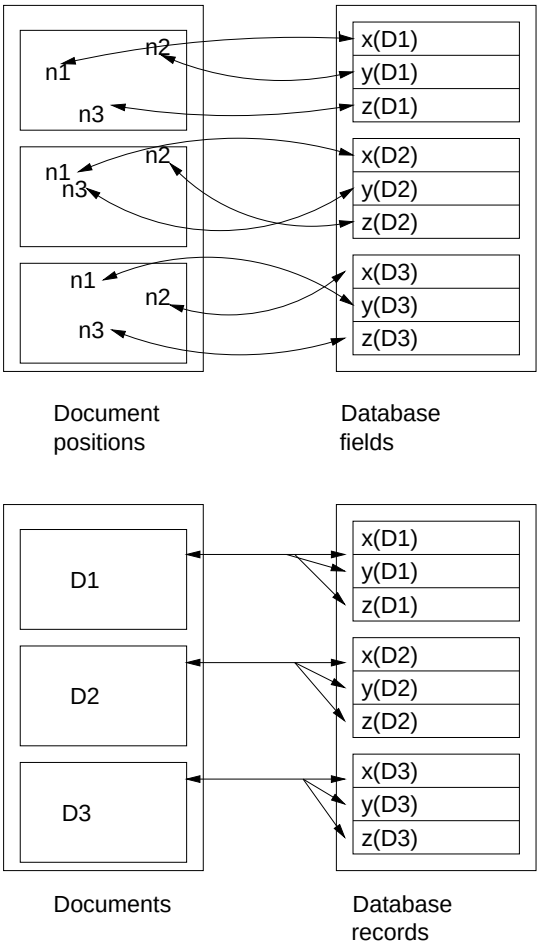

Figure 1: The ideal situation (above) for learning Information Extraction patterns, where each named entity n_i is marked in its context in a document and can be mapped to an associated database field, versus the commonly occurring situation (below) where each document is only globally associated with a database record.

methods that were used to select the features. Then, in Section 5. we give the accuracy results of the system in its final configuration, and attempt to draw some more general conclusions from this work.

2. Data

The data set on which we conducted this research is a corpus of 53645 job advertisements collected from the 1996-1999 archive of the Dutch online job search site **Intermediair Online** [1]. These advertisements have been scanned from printed editions and processed manually to correct OCR errors and add mark-up and classification information. The mark-up is in an SGML-like format. For our supervised learning approach the data presented two problems. First, about 20% of the corpus did not have classification labels. Second, many advertisements contained more than one position, which would confuse our classification-based approach. Therefore, we filtered the original dataset and obtained a set of 32442 documents

[1] http://www.intermediair.nl

```
<begin>
<lang>dutch
</lang>
<docid>374
</docid>
<paginanr> Computable 23/05/97 pagina  53
</paginanr>
<functie>
<utt>
systeem specialist rs/6000
</utt>
</functie>
<vaccode> 616515
<target>
<omschr>systeem specialist RS/6000
</omschr>
<funcode>C12
</funcode>
<funomschr>systeembeheerder
</funomschr>
<oplcode>G30
</oplcode>
<oplomschr>Informatica
</oplomschr>
<branchecode>8483
</branchecode>
<brancheomschr>Arbeidsbemiddelingsbureau (excl. overheid)
</brancheomschr>
</target>
</vaccode>
<empty val=2>
<arbeidsvoorwaarden>
<utt>
* spilfunctie in een sterk expanderende organisatie * werken in de
dynamische wereld van de financile dienstverlening * sterke
uitbreiding van activiteiten , organisatie en IT systemen *
geavanceerde bancaire systemen en infrastructuur * rele
persoonlijke groeikansen * uitstekende arbeidsvoorwaarden
</utt>
</arbeidsvoorwaarden>
<empty val=4>
<functieomschrijving>
<utt>
* technisch verantwoordelijk voor de RS/6000 systemen * technische
inrichting van de systeemomgeving * verantwoordelijk voor maximale
performance * connectivity en interfacing * inbrengen technische
kennis bij projecten * initiren en uitvoeren technische projecten
</utt>
</functieomschrijving>
<empty val=3>
<functieeisen>
<utt>
* HBO niveau * relevante IT - en producttrainingen * enige jaren
ervaring als Systeem Specialist RS/6000 * ervaring met RS/6000 multi
processor systemen is een pr * AIX * technische inrichting
systeemomgeving * installeren , configureren en uitvoeren technische
tests * kennis van netwerken en connectivity * Engelse taal *
technische affiniteit * zelfstandig * verantwoordelijkheidsgevoel *
flexibel * inzet * communicatief vaardig * ambitieus
</utt>
</functieeisen>
<empty val=3>
<aanvullend>
<utt>
Bel 0346 - 586000 voor uitwisseling van informatie !
</utt>
<utt>
Vraag naar de heer A.H.W. ( Dolf ) Kasteleijn .
</utt>
<utt>
Haselhoff Groep - Parkweg 53 - 3603 AB Maarssen Telefoon 0346 - 586000
- Fax 0346 - 565515 - E-mail : haselhof@pi.net Haselhoff Groep is
gespecialiseerd in de Werving \&amp;amp; Selectie voor de
lnformatica/telematica markt .
</utt>
</aanvullend>
</begin>
```

Figure 2: An example of a job advertisement document after all of our pre-processing steps.

which had exactly one label for all of the fields of interest. In the experiments reported below, we take the first 30000 documents as the training set and the remainder as the test set.

These documents were tokenized and split into sentences, and the language of the document was guessed using a freely available language guesser [2]. Figure 2 shows an example text after the preprocessing stage.

The documents are labeled with three information fields: job title code (henceforth `<funcode>`: 164 categories, e.g. “system administrator”, “commercial manager”, “accountant” etc.), education requirement code (henceforth

[2] Written by Gertjan van Noord, and available from http://odur.let.rug.nl/~vannoord/TextCat/. The data set contains about 5% of English texts, small portions are in German or French, and the large majority is Dutch

<oplcode>: 105 categories, e.g. “University level Computer Science”, “Law”, “Medicine”, “Social Sciences”, etc.), and Industry Sector code (henceforth <bracode>: 563 categories, e.g. “Publishing”, “Oil and coal producers”, “Breweries”, etc.). Because the data set consists of legacy data, the annotation has evolved and changed over the course of the years. Although we have no exact figures, anecdotal reports suggest that inter-annotator consistency is only about 60%. Moreover, many of the frequently assigned labels in all three fields are codes for non-specific or remainder categories. Therefore, the scores, which will be reported as accuracy of exact match with the human annotation of the test set may seem overly low.

In addition to the category labels, the texts are segmented into sections, and each section has a label from the following set: job title <functie>, description <functieomschrijving>, requirements <functieeisen>, organization description <bedrijfsomschrijving>, benefits <arbeidsvoorwaarden>, and contact information <aanvullend>. These labels are also to be predicted for a new text, given a correct segmentation of the text into sections (separated by whitespace). During the development of the system it became clear that it is useful to first label the sections, and then use this information to constrain the features for the fields of interest. For example “human resource manager” is not a good predictor of the position when present in the contact information section, but it is a very good feature when present in the job title. One of the innovative aspects of our system is that we also treat the problem of section labeling as a text classification task. Whereas typical features for the extraction of categories turn out to be content phrases (such as e.g. “managing director” or “systems programmer” etc.), the section labeling features are more syntactic filler phrases (e.g. “please contact”, “dynamic and growing”, “years of experience” etc.).

3. Memory-Based classification

To learn all of the classification tasks from labeled examples, we use a standard k-NN or Memory-Based Learning method (see (Daelemans et al., 2000) for a detailed description of an efficient Memory-Based Learning implementation). The examples are stored in memory as binary vectors [3]. For classification of a new case, its nearest neighbors are retrieved from memory, and the new case receives the most frequent category in the nearest neighbor set, or the distribution of categories in it. The distance metric that turned out to be very robust across all classification tasks is a simple overlap between two binary vectors, with all features receiving equal weight. (See e.g. (Yang and Pedersen, 1997; Creecy et al., 1992; Stanfill and Waltz, 1986) for earlier applications of Memory-Based Learning for Text Classification).

[3]We only store a sparse representation of the active features for a given case (in contrast to the implementation in (Daelemans et al., 2000)), and use an inverse index from features to cases to speed up the matching process.

4. Feature Mining

The size of the category set in our application makes it necessary to use a very large feature set, because in order to make generalizations, we need to recognize many different (string realizations of) cues for each category. Our feature language allows the following types of features:

- unigrams: a string not interrupted by whitespace.
- phrase: a string consisting of several consecutive unigrams separated by whitespace.
- combi: a conjunction of a unigram or phrase and a specific section label.
- regexp: any regular expression on the text (these are not used in the experiments described in this paper).

A feature is one (active) if it is present in a text and zero otherwise. To represent a document we generate a vector of all of its active features. However, not all potential features are actually selected. For both computational reasons and successful generalization, we need to constrain the document representations to contain only features that are good predictors. For example, stop words are not very useful for discriminating among classes and will hide other “good” features; and once-occurring strings have a low chance of generalization to new documents and are thus only a burden on space.

Each unique token occurring the in the training corpus is a potential feature; a total of more than 170 thousand candidates for our data set. We have done a comparison of several well-known corpus based measures for evaluating the predictiveness of single term features: Gain Ratio (GR) (Quinlan, 1993), Information Gain (IG), Chi Squared (CHI) (Yang and Pedersen, 1997), log likelihood (LL) (Dunning, 1993), raw frequency (FREQ), and a measure of class stickiness (maximum conditional probability of a class given a feature, combined with an absolute frequency threshold[4] (CP) (Ng and Lee, 1996).

Because single term features are not precise enough for our task, we need phrase features as well. For example, the terms ”network” and ”engineer” by themselves are only mediocre predictors, but ”network engineer” is a very specific cue for the job title. However, for phrasal features, we cannot even consider to compute the selection statistics for all possible phrases as their number quickly explodes. To mine for phrasal features, we performed ranking of all n-grams up to length twelve in the corpus by their log-likelihood (Dunning, 1993). Then we applied a threshold that left us with about a million collocation-like n-grams. These n-grams were then again ranked by their predictive quality for the three content entities and the section labeling, using the CP measure.

To allow more precise matching of phrases, our system first labels the sections, and then considers conjunctions of phrases and section labels as features. These conjunctions are again only used if they fall above a certain threshold of predictiveness.

[4]The threshold was set to be 5.

The system described so far is constructed in a completely automatic way from the training data. A final addition to the system was to mine the corpus semi-automatically for particular types of phrases, viz. those describing jobs, educations, and companies. For this we used methods similar to those described by (Brin, 1998), starting from a very small seed set of known entities, finding the contexts these occur in, and under human supervision iteratively identifying entities and contexts from there. Moreover, several pre-existing lists of company education and job names were included as features as well.

5. Results

5.1. Feature selection

Table 1 shows the results of experiments with the aforementioned statistical feature selection measures, applied to unigram (single term) features.

The main conclusion from this table is that more features is almost always better. In fact, in the final system we have included many more features than the maximum of 10000 in this table.

For the education field the simple metric CP outperforms all others. For job title, log likelihood (LL) beats it by a small margin. On the sector field, IG, GR and FREQ turned out to perform best. What is surprising is that simple measures, such as CP and FREQ are doing so well, in comparison with the more sophisticated measures. Our explanation for this is that for these large numbers of categories, statistical measures such as IG and CHI, which are measuring the deviation of the conditional probability $P(c|feature)$ from the a priori probability $P(c)$ over all categories c, are misled by a large number of small deviations. Thus they tend to select less informative frequent features over more informative infrequent features. In previous work in text classification, the examination of feature selection methods has mostly confined itself to binary classification, where this problem is much smaller. Gain Ratio, on the other hand, tries to correct for the frequency of the feature, but overestimates the importance of infrequent features, so that at 10000 features, about 90% of the cases still have zero active features. In contrast, with IG, CHI and FREQ 100% of the cases have at least one active feature, and with CP about 90%. For a more detailed report of explorations of the parameter space see (Lavrijssen, 2000)).

Although the resulting system does not completely solve the extraction task for job ads, it goes quite a long way, given the short development period. Using the finalized sets of features, the accuracy of the section labeling is quite high: 87%. Extracting and categorizing positions has 50%, education 56% and industry sector 74% accuracy. However, these figures are measured on a rather noisy corpus of legacy data. If we relax the class boundaries a bit, and score all ”plausible” answers as correct (effectively merging some related labels), the results improve to respectively 76%, 59% and 81% for job title, education, and sector. Using these classifiers the system is embedded in a web-based interface (shown in Figure 3). In this interface, the suggestions made by the system can efficiently be corrected by human operators. Moreover, the case-based nature of the system allows for continuous incremental updating of both the feature set and the memory of classified cases, so that the system can further be improved by applying it.

Although using a text classification approach might seem counterintuitive for the solution of an IE task, we have shown that it can be applied successfully in a limited domain such as job advertisements, if detailed named entity labeling is not available for training.

Acknowledgements

The construction of the Textractor system would not have been possible without the dedication, support and hard work of our colleagues of the ILK group at Tilburg University (in alphabetical order): Antal van den Bosch, Sabine Buchholz, Walter Daelemans, and Jorn Veenstra. Furthermore the authors would like to thank Peter Went and Wouter Went of WCC Services BV. for their involvement in the project, and Paul Sweere of VNU Business Publications for providing us with the job advertisement data.

6. References

Brin, S., 1998. Extracting patterns and relations from the world wide web. In *Proceedings of WebDB Workshop at EDBT'98, Valencia, Spain, March 1998*.

Califf, M.E. and R.J. Mooney, 1999. Relational learning of pattern-match rules for information extraction. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI'99)*.

Creecy, R.H., B.M. Masand, S.J. Smith, and D.J. Waltz, 1992. Trading MIPS and memory for knowledge engineering. *Communications of the ACM*, 35:48–64.

Daelemans, W., J. Zavrel, K. van der Sloot, and A. van den Bosch, 2000. TiMBL: Tilburg memory based learner, version 3.0, reference manual, technical report ILK-0001. Technical report, ILK, Tilburg University.

Dunning, T., 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1).

Freitag, D., 1998. Information extraction from HTML. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI'98)*.

Lavrijssen, W., 2000. Stageverslag: Feature-selectie voor automatische informatie-extractie uit vacatureteksten (in Dutch), Tilburg University. Undergraduate Research Report.

Mladenic, D., 1998. Feature subset selection in text-learning. In *Proceedings of the 10th European Conference on Machine Learning ECML98*. Springer Verlag, Berlin.

Mladenic, D. and M.D. Grobelnik, 1998. Word sequences as features in text-learning. In *of the Seventh Electrotechnical and Computer Sc. Conference ERK'98*.

Ng, H.T. and H.B. Lee, 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th Annual Meeting of ACL*. Morgan Kaufmann, San Mateo.

Quinlan, J.R., 1993. C4.5*: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

		Accuracy %				
field	selection method	100	500	2000	5000	10000
opleiding (education)	IG	28.82	29.72	32.38	33.36	34.06
	GR	5.08	5.32	6.26	6.84	8.39
	CHI	18.91	34.34	32.50	32.50	32.05
	FREQ	15.76	27.67	31.93	33.12	33.97
	CP	0.82	3.32	13.30	31.76	**41.55**
	LL	33.65	36.84	38.97	38.56	37.29
functie (job title)	IG	20.67	26.48	28.12	29.31	29.84
	GR	0.00	0.33	5.61	7.41	9.74
	CHI	12.81	25.01	28.08	29.02	29.39
	FREQ	12.32	24.48	27.83	29.14	29.88
	CP	1.15	4.26	16.82	26.03	35.33
	LL	24.68	34.59	**35.41**	33.73	33.44
branche (sector)	IG	38.19	53.99	59.35	62.46	63.57
	GR	0.12	0.29	2.09	3.77	15.02
	CHI	0.12	0.29	13.75	26.36	49.24
	FREQ	22.35	50.59	59.72	62.63	63.28
	CP	1.80	7.20	17.81	40.24	53.54
	LL	28.98	56.04	61.36	63.04	**63.61**

Table 1: Comparison of several feature-selection measures on single term (unigram) features. The table shows the accuracy percentage of the most likely category for three tasks when trained on 30 thousand documents, and tested on the remaining 2442 documents. For each selection method a row shows the results if the n top ranked terms are used as features.

Soderland, S., 1999. Learning information extraction rules from semi-structured and free text. *Machine Learning*, 34:233–272.

Spitters, M., 1999. *Automatic Text Categorization, an experimental study of feature selection methods and machine learning techniques for the automatic categorization of Dutch newspaper-articles*. Master's thesis, Tilburg University.

Stanfill, C. and D. Waltz, 1986. Toward memory-based reasoning. *Communications of the* ACM, 29(12):1213–1228.

Yang, Y. and J.O. Pedersen, 1997. A comparative study on feature selection in text categorization. In *Proc. of the 14th International Conference on Machine Learning ICML97*.

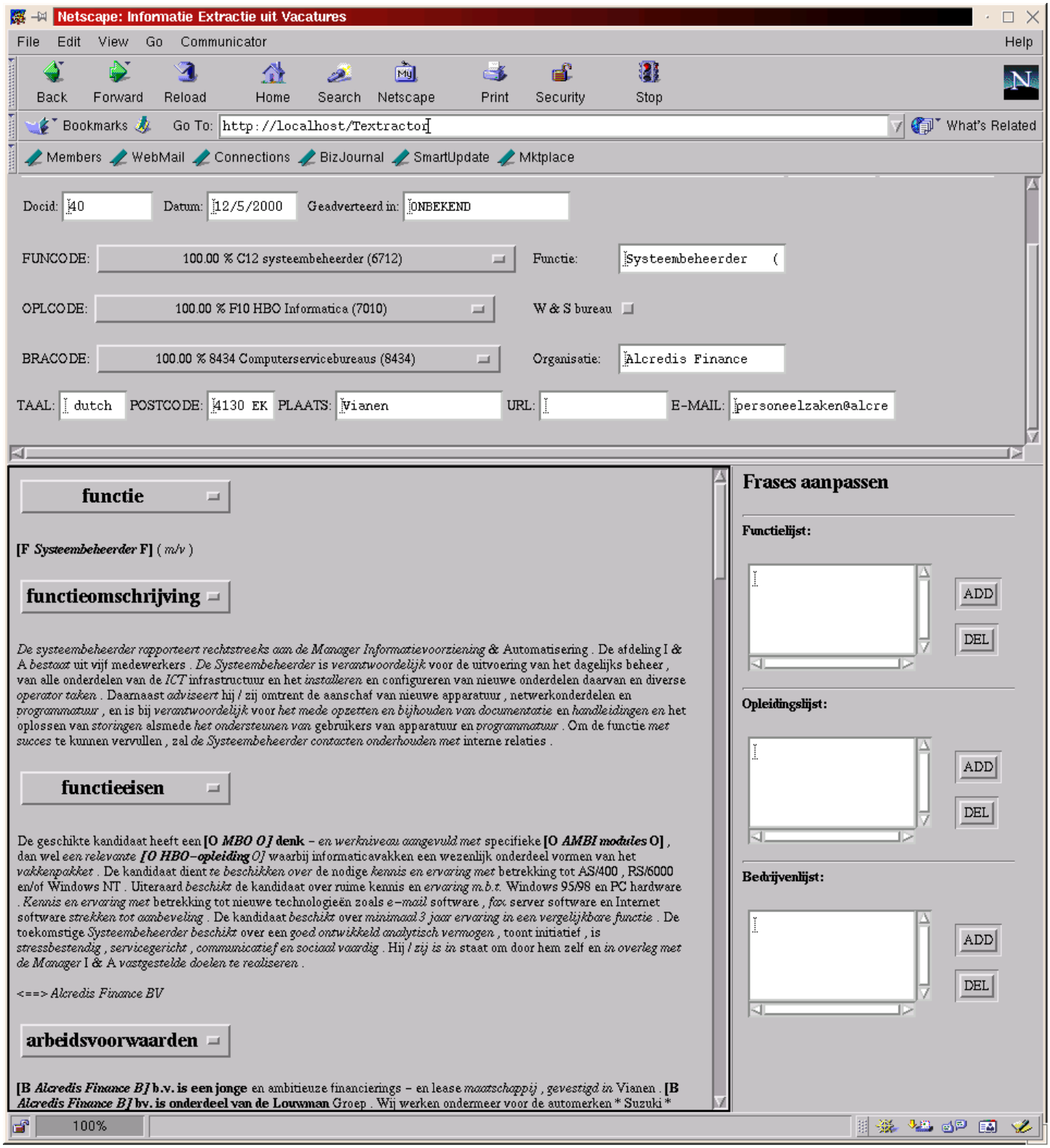

Figure 3: The interface for the Textractor system.