Classifying Number Expressions in German Corpora

Irene Cramer[1], Stefan Schacht[2], and Andreas Merkel[2]

[1] Dortmund University irene.cramer@uni-dortmund.de
[2] Saarland University firstname.lastname@lsv.uni-saarland.de

Abstract. Number and date expressions are essential information items in corpora and therefore play a major role in various text mining applications. However, so far number expressions were investigated in a rather superficial manner. In this paper we introduce a comprehensive number classification and present promising, initial results of a classification experiment using various Machine Learning algorithms (amongst others AdaBoost and Maximum Entropy) to extract and classify number expressions in a German newspaper corpus.

1 Introduction

In many natural language processing (NLP) applications such as Information Extraction and Question Answering number expressions play a major role, e.g. questions about the altitude of a mountain, the final score of a football match, or the opening hours of a museum make up a significant amount of the users' information need. However, common Named Entity task definitions do not consider number and date/time expressions in detail (or as in the Conference on Computational Natural Language Learning (CoNLL) 2003 (Tjong Kim Sang (2003) do not incorporate them at all). We therefore present a novel, extended classification scheme for number expressions, which covers all Message Understanding Conference (MUC) (Chinchor (1998a)) types but additionally includes various structures not considered in common Named Entity definitions. In our approach, numbers are classified according to two aspects: their function in the sentence and their internal structure. We argue that our classification covers most of the number expressions occurring in text corpora. Based on this classification scheme we have annotated the German CoNLL 2003 data and trained various machine learning algorithms to automatically extract and classify number expressions. We also plan to incorporate the number extraction and classification system described in this paper into an open domain Web-based Question Answering system for German. As mentioned above, the recognition of certain date, time, and number

expressions is especially important in the context of Information Extraction and Question Answering. E. g. the MUC Named Entity definitions (Chinchor (1998b)) include the following basic types: date, time (<TIMEX>) as well as monetary amount and percentage (<NUMEX>), and thus fostered the development of extraction systems able to handle number and date/time expressions. Famous Information Extraction systems developed in conjunction with MUC are e.g. FASTUS (Appelt et al. (1993)) or LaSIE (Humphreys et al. (1998)). At that time, many researchers used finite-state approaches to extract Named Entities. More recent Named Entity definitions, such as CoNLL 2003 (Tjong Kim Sang (2003)), aiming at the development of Machine Learning based systems, however, again excluded number and date expressions. Nevertheless, due to the increasing interest in Question Answering and the TREC QA tracks (Voorhees et al. (2000)), recently, a number of research groups investigate various techniques to fast and accurately extract information items of different types form text corpora and the Web, respectively. Many answer typologies naturally include number and date expressions, e.g. the ISI Question Answer Typology (Hovy et al. (2002)). Unfortunately, in the corresponding papers only the whole Question Answering System's performance is specified, we therefore could not detect any performance values, which would be directly comparable to our results. A very interesting and partially comparable (they only consider a small fraction of our classification) work (Ahn et al. (2005)) investigates the extraction and interpretation of time expressions. Their reported accuracy values range between about 40% and 75%.

Paper Plan: This paper is structured as follows. Section 2 presents our classification scheme and the annotation. Section 3 deals with the features and the experimental setting. Section 4 analyzes the results and comments on the future perspectives.

2 Classification of Number Expressions

Many researchers use regular expressions to find numbers in corpora, however, most numbers are part of a larger construct such as 2,000 miles or Paragraph 249 Bürgerliches Gesetzbuch. Consequently, the number without its context has no meaning or is highly ambiguous (2,000 miles vs. 2,000 cars). In applications such as Question Answering it is therefore necessary to detect this additional information. Table 1 shows example questions that obviously ask for number expressions as answers. The examples clearly indicate that we are not looking for mere digits but multi-word units or even phrases consisting of a number and its specifying context. Thus, a number is not a stand-alone information and, as the examples show, might not even look like a number at all. This paper therefore proposes a novel, extended classification that handles number expressions similar to Named Entities and thus provides a flexible and scalable method to incorporate these various entity types into one generic framework. We classify numbers according to their internal structure (which

corresponds to their text extension) and their function (which corresponds to their class).

Table 1. Example Questions and Corresponding Types

Q: `How far is the Earth from Mars?`	miles? light-years?
Q: `How high is building X?`	meters? floors?
Q: `What are the opening hours of museum X?`	daily from 9 am to 5 pm
Q: `How did Dortmund score against Cottbus last weekend?`	2:3

We also included all MUC types to guarantee that our classification conforms with previous work.

2.1 Classification Scheme

Based on Web data and a small fraction of online available German newspaper corpora (Frankfurter Rundschau[1] and die tageszeitung[2]) we deduced 5 basic types: `date` (including date and time expressions), `number` (covering count and measure expressions), `itemization` (rank and score), `formula`, and `isPartofNE` (such as street number or zipcode). As further analyses of the corpora showed most of the basic types naturally split into sub-types, which also conforms to the requirements imposed on the classification by our applications. The final classification thus comprises the 30 classes shown in table 2. The table additionally gives various examples and a short explanation of the class' sense and extension.

2.2 Corpora and Annotation

According to our findings in Web data and newspaper corpora we developed guidelines which we used to annotate the German CoNLL 2003 data. To ensure a consistent and accurate annotation of the corpus, we worked every part over in several passes and performed a special reviewing process for critical cases. Table 3 shows an exemplary extract of the data. It is structured as follows: the first column represents the token, the second column its corresponding lemma and the third column its part-of-speech, the fourth column specifies the information produced by a chunker. We did not change any of these columns. In column five, typically representing the Named Entity tag, we added our own annotation. We replaced the given tag if we found the tag `O` (`=other`) and appended our classification in all other cases.[3] While annotating the corpora we met a number of challenges:

- **Preprocessing:** The CoNLL 2003 corpus exhibits a couple of erroneous sentence and token boundaries. In fact, this is much more problematic for

[1] http://www.fr-online.de/

[2] http://www.taz.de/

[3] Our annotation is freely available for download. However, we cannot provide the original CoNLL 2003 data, which you need to reconstruct our annotation.

the extraction of number expressions than for Named Entity Recognition, which is not surprising, since it inherently occurs more frequently in the context of numbers.

Table 2. Overview of Number Classes

Name of Sub-Type	Examples	Explanation
`date.period`	for 3 hours, two decades	time/date period, start and end point not specified
`date.regular`	weekdays 10 am to 6 pm	expressions like opening hours etc.
`date.time`	at around 11 o'clock	common time expressions
`date.time.period`	6-10 am	duration, start and end specified
`date.time.relative`	in two hours	relative specification tie: e.g. now
`date.time.complete`	17:40:34	time stamp
`date.date`	October 5	common date expressions
`date.date.period`	November 22-29, Wednesday to Friday, 1998/1990	duration, start and end specified
`date.date.relative`	next month, in three days	relative specification tie: e.g. today
`date.date.complete`	July 21, 1991	complete date
`date.date.day`	on Monday	all weekdays
`date.date.month`	last November	all months
`date.date.year`	1993	year specification
`number.amount`	4 books, several thousand spectators	count, number of items
`number.amount.age`	aged twenty, Peter (27)	age
`number.amount.money`	1 Mio Euros, 1,40	monetary amount
`number.amount.complex`	40 children per year	complex counts
`number.measure`	18 degrees Celsius	measurements not covered otherwise
`number.measure.area`	30.000 acres	specification of area
`number.measure.speed`	30 mph	specification of speed
`number.measure.length`	100 km bee-line, 10 meters	specification of length, altitude, ...
`number.measure.volume`	43,7 l of rainfall, 230.000 cubic meters of water	specification of capacity
`number.measure.weight`	52 kg sterling silver, 3600 barrel	specification of weight
`number.measure.complex`	67 l per square mile, 30x90x45 cm	complex measurement
`number.percent`	32 %, 50 to 60 percent	percentage
`number.phone`	069-848436	phone number
`itemization.rank`	third rank	ranking e.g. in competition
`itemization.score`	9 points, 23:26 goals	score e.g. in tournament
`formula.variables`	$\prod \cos(x)$	generic equations
`formula.parameters`	$y = 4.132 * x^3$	specific equations

- **Very complex expressions:** We found many `date.relative` and `date.regular` expressions, which are extremely complex types in terms of length, internal structure, as well as possible phrasing and therefore difficult to extract and classify. In addition, we also observed very complex `number.amount` contexts and a couple of broken sports score tables, which we found very difficult to annotate.
- **Ambiguities:** In some cases we needed a very large context window to disambiguate the expressions they annotated. Additionally, we even found

examples which we could not disambiguate at all. E.g. *über 3 Jahre* with the possible translations *more than 3 years* or *for 3 year*. In German such structures are typically disambiguated by prosody.

- **Particular text type:** A comparison between CoNLL and the corpora we used to develop our guidelines showed that there might be a very particular style. We also had the impression that the CoNLL training and test data differ with respect to type distribution and style. We therefore based our experiments on the complete data and performed cross-validation.

We think that the thus annotated corpora represent a valuable resource, especially, given the well-known data sparseness for German.

Table 3. Extract of the Annotated CoNLL 2003 Data

Am	am	APPRART	I-PC	date.date.complete
14.	14.	ADJA	I-NC	date.date.complete
August	August	NN	I-NC	date.date.complete
1922	@card@	CARD	I-NC	date.date.complete
rief	rufen	VVFIN	I-VC	O
er	er	PPER	I-NC	O
den	d	ART	B-NC	O
katholischen	katholisch	ADJA	I-NC	O
Gesellenverein	Gesellenverein	NN	I-NC	O
ins	ins	APPRART	I-PC	O
Leben	Leben	NN	I-NC	O
.	.	$.	O	O

Furthermore, our findings during the annotation process again emphasized the need of an integrated concept of number expressions and Named Entities: we found 467 `isPartofNE` items, which are extremely difficult to classify without any hint about proper names in the context window.

3 Experimental Evaluation

3.1 Features

Our features (see table 4 for details) are adapted from those reported in previous work on Named Entity Recognition (e.g. Bikel et al. (1997), Carreras et al. (2003)). We based the extraction on a very simple and fast analysis of the tokens combined with shallow grammatical clues. To additionally capture information about the context we used a sliding window of five tokens (the word itself, the previous two, the following two).

3.2 Classifiers

To get a feeling for the expectable performance, we conducted a preliminary test by experimenting with Weka (Witten et al. (2005)). For this purpose we ran the Weka implementations of a Decision Tree, k-Nearest Neighbor, and Naive Bayes algorithm with the standard settings and no preprocessing or tuning. Because of previous, promising experiences with AdaBoost (Carreras

et al. (2003)) and Maximum Entropy in similar tasks, we decided to also apply these two classifiers. We used the `maxent` implementation of the Maximum Entropy algorithm[4]. For the experiments with AdaBoost we used our own `C++` implementation, which we tuned for large sparse feature vectors with binary entries.

Table 4. Overview of Features Used

feature group	features
only digit strings	2-digit integer [30-99], other 2-digit integer, 4-digit integer [1000-2100], other 4-digit integer, other integer
digit and non-digit strings	1-digit or 2-digit followed by point, 4-digit with central point or colon, any digit sequence with point, colon, comma, comma and point, hyphen, slash, or other non-digit character
non-digit strings	any character sequence max length 3, any character sequence, followed by point, any character sequence with slash, any character sequence
grammar	part-of-speech tag, lemma
window	all features mentioned above for window +/-2

3.3 Results

The performance of the Decision Tree, k-Nearest Neighbor, Naive Bayes, and Maximum Entropy algorithm is on average mediocre, as Table 5 reveals. On the contrary, our AdaBoost implementation shows satisfactory or even good f-measure values for almost all cases and thus significantly outperforms the rest of the classifiers.

Table 5. Overview of the F-Measure Values (AB: AdaBoost, DT: Decision Tree, KNN: k-Nearest Neighbor, ME: Maximum Entropy, NB: Naive Bayes)

class	AB	DT	KNN	ME	NB	class	AB	DT	KNN	ME	NB
other	0.99	0.99	0.98	0.99	0.97	itemization.score	**0.83**	0.43	0.40	0.78	0.04
date	**0.37**	0.13	0.21	0.24	0.19	number	**0.64**	0.00	0.08	0.00	0.00
date.date	0.67	0.73	0.67	**0.74**	0.09	number.amount	0.33	0.53	0.25	**0.67**	0.26
date.date.complete	0.72	0.61	**0.74**	0,49	0.20	number.amount.age	**0.62**	0.28	0.14	0.45	0.02
date.date.day	**0.53**	0.15	0.14	0.20	0.06	number.amount.complex	**0.09**	0.00	0.00	0.00	0.00
date.date.month	**0.37**	0.05	0.08	0.24	0.00	number.amount.money	**0.82**	0.45	0.28	0.79	0.30
date.date.period	**0.43**	0.38	0.36	0.45	0.09	number.measure	**0.22**	0.16	0.00	0.17	0.00
date.date.relative	0.54	0.36	0.16	**0.59**	0.00	number.measure.area	**0.88**	0.10	0.00	0.40	0.00
date.date.year	**0.82**	0.73	0.58	0.76	0.60	number.measure.complex	**0.34**	0.21	0.19	0.22	0.09
date.regular	0.49	0.43	0.37	**0.54**	0.14	number.measure.length	**0.69**	0.17	0.11	0.39	0.01
date.time	**0.87**	0.76	0.67	0.83	0.45	number.measure.speed	**0.91**	0.17	0.18	0.00	0.00
date.time.period	0.41	0.40	**0.46**	0.38	0.31	number.measure.volume	**0.66**	0.06	0.00	0.00	0.00
date.time.relative	**0.38**	0.02	0.07	0.00	0.00	number.measure.weight	**0.49**	0.00	0.00	0.00	0.00
itemization	0.21	**0.28**	0.23	0.17	0.12	number.percent	**0.83**	0.32	0.10	0.56	0.06
itemization.rank	**0.84**	0.31	0.23	0.70	0.00	number.phone	**0.96**	0.85	0.89	0.95	0.65

Table 5 also shows that there are classes with a consistently poor performance, such as `number.amount.complex`, `number. measure`, or `itemization`, and a consistently good performance, such as `number.phone` or `date.date.year`. We think that this correlates with the amount of data as well as the heterogeneity of the classes. For instance, `number.measure` and `itemization` items

[4] http://www2.nict.go.jp/x/x161/members/mutiyama/software.html

occur indeed frequently in the corpus but these two classes are–according to our definition–'garbage collectors' and therefore much less homogenous. In contrast, there are classes, such as `date.time.period` or `date.regular`, with rather low f-measure values but a very precise definition; we admittedly suspect that the annotation of these types in our corpora might be inconsistent or inaccurate. We also suppose that there are number expressions which exhibit an exceedingly large variety of phrasing. As a matter of fact, these are inherently difficult to learn if the data do not feature sufficient coverage.

Table 6. Overview of the Precision Values (AB: AdaBoost)

class	AB	class	AB	class	AB
other	0.98	date.time	0.88	number.amount.complex	0.39
date	0.61	date.time.period	0.54	number.percent	0.87
date.date	0.75	date.time.relative	0.50	number.phone	0.96
date.date.complete	0.79	itemization	0.34	number.measure	0.70
date.date.day	0.83	itemization.rank	0.88	number.measure.area	0.93
date.date.month	0.79	itemization.score	0.91	number.measure.length	0.85
date.date.year	0.85	number	0.81	number.measure.speed	0.94
date.date.relative	0.73	number.amount	0.48	number.measure.volume	0.76
date.date.period	0.65	number.amount.age	0.79	number.measure.weight	0.56
date.regular	0.68	number.amount.money	0.89	number.measure.complex	0.65

Fortunately, there are is a number of classes with a pretty high f-measure value–that is more than 0.8–for at least one of the five classifiers, e.g. `date.date.year`, `itemization.rank`, and `number.phone`. More importantly there are, as Table 6 shows, only six classes with a precision value of less than 0.6. We are therefore very confident to be able to successfully integrate the AdaBoost implementation of our number extraction component into a Web-based open domain Question Answering System, since in a Web-based framework the focus tends to be on precision rather than coverage or recall.

4 Conclusions and Future Work

We presented a novel, extended number classification and developed guidelines to annotate a German newspaper corpus accordingly. On the basis of our annotated data we have trained and tested five classification algorithms to automatically extract and classify them with promising evaluation results. However, the accuracy is still low for some classes, especially for the small or heterogenous ones. But we feel confident to improve our system by incorporating selected training data, especially, in the case of small classes. To find the weak points in our system, we plan to perform a detailed analysis of all number types and their precision, recall, and f-measure values. We also consider a revision of our annotation, because there still might be inconsistently and inaccurately annotated sections in the corpus. As mentioned above, the CoNLL 2003 data exhibit a typical newspaper style, which might limit the applicability of our system to particular corpus types (although, initial experiments with Web data do not support this skepticism). We therefore intend

to augment our training data with Web texts annotated according to our guidelines. In addition, we plan to experiment with an expanded feature set and several pre-processing methods such as feature selection and normalization. Research in the area of Named Entity extraction shows that multiple classifier systems or the concept of multi-view learning might be especially effective in our application. We therefore plan to investigate several classifier combinations and also take a hybrid approach—combining grammar rules and statistical methods—into account. We plan to integrate our number extraction system into a Web-based open domain Question Answering system for German and hope to improve the coverage and performance of the answer types processed. While there is still room for improvement, we think–considering the complexity of our task–the achieved performance is surprisingly good.

References

AHN, D. / FISSAHA ADAFRE, S. / DE RIJKE, M. (2005): Recognizing and Interpreting Temporal Expressions in Open Domain Texts. *S. Artemov et al. (eds): We Will Show Them: Essays in Honour of Dov Gabbay, Vol 1., College Publications.*

APPELT, D. / BEAR, J. / HOBBS, J. / ISRAEL, D. / KAMEYAMA, M. / STICKEL, M. / TYSON, M. (1993): FASTUS: A Cascaded Finite-State Tranducer for Extracting Information from Natural-Language Text. *SRI International.*

BIKEL, D. / Miller, S. / Schwartz, R. / Weischedel, Ralph (1997): Nymble: a high-performance learning name-finder. *Proceedings of 5th ANLP.*

CARRERAS, X. / Mrquez, L. / Padr, L. (2003): A Simple Named Entity Extractor using AdaBoost. *Proceedings of CoNLL-2003*

CHINCHOR, Nancy A. (1998a): Overview of MUC-7/MET-2. *Proceedings of the Message Understanding Conference 7.*

CHINCHOR, N. A. (1998b): MUC-7 Named Entity Task Definition (version 3.5) *Proceedings of the Message Understanding Conference 7.*

HOVY, E. H. / HERMJAKOB, U. / RAVICHANDRAN, D. (2002): A Question/Answer Typology with Surface Text Patterns. *Proceedings of the DARPA Human Language Technology conference (HLT).*

HUMPHREYS, K. / GAIZAUSKAS, R. / AZZAM, S. / HUYCK, C. / MITCHELL, B. / CUNNINGHAM, H. / WILKS, Y. (1998): University of Sheffield: Description of the LaSIE-II System as Used for MUC-7. *Proceedings of the 7th Message Understanding Conference (MUC-7).*

TJONG KIM SANG, E. F. / DE MEULDER, F. (2003): Introduction to the CoNLL Shared Task: Language-Independent Named Entity Recognition. *Proceedings of the Conference on Computational Natural Language Learning.*

VOORHEES, E. / TICE, D. (2000): Building a Question Answering Test Collection. *Proceedings of SIGIR-2000.*

WITTEN, I. H. / FRANK, E. (2005): *Data Mining: Practical machine learning tools and techniques.* 2nd Edition, Morgan Kaufmann, San Francisco.