Using Markov Models for Named Entity recognition in German newspapers

Marc Rössler
Gerhard-Mercator University of Duisburg
Computational Linguistics
Geibelstraße 41
D-47048 Duisburg
marc.roessler@uni-duisburg.de

Abstract

This paper describes preliminary experiments for a system of named entity recognition in German newspapers. The approach is based on second order Markov Models trained on a tagged corpus. No gazetteers are used, only a list of words providing evidence is integrated. These words are extracted by statistical methods from an annotated corpus. The input basically consists of a part of speech tagged text, except the words occurring in the gained list, which replace the tags with their word form. The experiments investigate in how far such a limited approach is suitable for German and show that it provides some evidence. However, of course, it has to be enhanced.

1 Introduction

The recognition of named entities (NE) deals with the detection and categorization of proper names. Which elements count as NE and which factors define these categories is not always clear and depends on the application. Frequently used categories are persons, companies and / or organizations, locations and sometimes temporal and numeric expressions. The resulting data can be utilized for information extraction, information retrieval, topic detection, summarization and other natural language applications. Various methods are applied, ranging from very large lists to different statistical and machine learning approaches.

If large lists containing proper names are available, the usage of their items is especially suitable for NE which occur often. Nevertheless, there will always be a lack of coverage, especially in lists of companies, which tend to be out of date rapidly, since new companies with new names are founded very often. However, also lists of person names have to be very large and are still far from being complete.

A second aspect of the task concerns the variation of NE. Since all occurrences referring to a particular category have to be found, it is not sufficient to detect, for example, "IBM", as this company is also named "International Business Machines Corporation" and "Big Blue".

The task is especially difficult in German, because not only proper nouns, but also regular ones are capitalized in German. Additionally, adjectives derived from geographical names are only capitalized when they end with "er". As an example the adjective "deutsch" (engl. "German") starts with a lower case in contrast to "Schweizer" (engl. "Swiss"). This means that capitalization, an important feature for most systems of NE recognition, is not that valuable in German.

This paper reports some preliminary experiments for NE recognition in German newspaper texts. The approach is based on second order Markov Models trained on a tagged corpus and does not use any gazetteers. Therefore, the question to answer is: In how far is an approach which is limited to a trigram window and does not exploit additional knowledge sources, applicable for the German language?

The *Computer Zeitung*, a German computer magazine, was selected as research corpus. It appears weekly and all issues from 1993 to 1996 (CZ 1998) are available on CD-ROM.

2 Approach

When inspecting texts for the purpose of identifying NE, it is obvious that there are some words which can indicate the occurrence of a NE. According to McDonald (1996), there are two complementary kinds of evidence: *Internal* evidence is taken from within the NE, while *external* evidence is provided by the context in which a name appears.

For some words it is nearly certain that they are a part of, followed or preceded by a particular NE. For other words, this is only sometimes the case. Examples of the former are "Mr" or "Prof.", followed by a proper name. Examples of the latter are company names followed by "takes over" or "produces". Rule based systems try to utilize this phenomenon in two steps: First, words indicating a NE are collected. Second, rules are written taking the occurrence of others words as well as other evidence into account, such as part of speech, the appearance of upper and lower case letters etc. Combined with additional heuristics, gazetteers or other lists containing proper names, such an approach produces good results (e.g. Volk and Clematide, 2001), but requires a great deal of laborious work. Furthermore, relying on available lists is only possible, if they are kept continuously up to date.

The basic idea of our approach is the use of Markov Models as a "lazy" method of pattern matching identification. "Lazy" in this respect means that in the prototype neither handcrafted lists of words indicating NE nor any handcrafted rules are employed.

Markov Models are finite state automata with probabilistic state transitions and symbol emissions. They are defined by the probability of the state transitions and the probability of emitting an output symbol by a particular state. TnT (Brants, 1998, 2000), the implementation of the Markov Model used, was originally designed for part of speech tagging. It is a second order Markov model, which means that the transition probabilities are calculated considering two instead of only one preceeding state. TnT generates probabilities from a tagged corpus and stores them in a lexicon and an n-gram file. The lexicon file contains the frequencies of the token and the tags it occurred with. The n-gram file contains the contextual frequencies for uni-, bi- and trigrams. Due to the sparse data problem, the generated trigram probabilities cannot be used directly. Therefore, the n-gram probabilities are smoothed by linear interpolation.

The design of our system's input was motivated by dealing with the sparse data problem as well as by providing internal and external evidence. Therefore, the input basically consists of part of speech tags except the words that seemed to bear evidence for the occurrence of a NE. With this input we intended that the model learns some of the specific patterns of the part of speech tags within and surrounding a NE. At the same time, it was supposed to consider words indicating a certain NE category.

Three categories of NE were to be recognized: PERSON, ORGANIZATION and LOCATION. In the beginning, we attempted to work with four different categories of NE: PERSON, LOCATION, COMPANY and ORGANIZATION. However, the distinction between COMPANY and ORGANIZATION, which should distinguish profit from non-profit organizations, was given up, since the annotation was often vague. Moreover, the first results were disappointing for the detection of ORGANIZATION. So these two categories were subsumed under ORGANIZATION.

Previous research on NE recognition mainly focused on English. Approaches using learning algorithms like Hidden Markov Models (e.g BBN's IdentiFinder, Bikel et al. 1999; most recent Zhou and Su, 2002) or Maximum Entropy Modelling (e.g. MEME, Borthwick et al. 1998) were among the best performing at the MUC-7 (Message Understanding Conference 1998). Mikheev et al. work with a hybrid approach combining rules with maximum entropy modelling (Mikheev et al. 1998). They also report experiments on the impact of gazetteers, running their system with a full, a limited and without any gazetteer (Mikheev et al. 1999).

Systems for NE recognition in German are rare. Volk and Clematide (2001) used precompiled lists for locations and persons and miscellaneous patterns to detect company names. They report a precision of 92% and a recall of 86% for person names. Companies are recognized with a precision of 76% and a recall of 81%, locations with a precision of 81% and a recall of 91%. Neumann and Piskorski (2000) describe a system for intelligent text extraction which includes a NE recognition based on finite state machines and several knowledge sources. They report a precision of 95% and a recall of 85%, but it is not clear whether they describe the detection of NE in contrast to regular nouns or the classification of NE.

3 The corpus annotation

The manually annotated corpus consists of about four issues of the *Computer Zeitung* (CZ 1998) from the year 1996. It contains about 100 000 tokens and was annotated by a student with a simple HTML editor within 20 hours and without any other annotation to check the results. The categories used to annotate were PERSON, ORGANIZATON and LOCATION.

Problems during the annotation concerned the above mentioned distinction between COMPANY and ORGANIZATION, which was difficult, often too vague and thus given up. The status of newspapers and online portals like "AOL" was not clear either: Does the name refer to the publishing company or to the publication? However, as this was not our actual topic, the problem was solved using annotation guidelines like "never annotate newspapers" or "always annotate online portals" etc. Table 1 shows the number of NE found in the corpus and the number of tokens they consisted of.

category	PERSON	ORGANIZATION	LOCATION
number of NE	824	2769	1450
number of tokens	1404	4249	1506

Table 1: Occurrence of NE in the annotated corpus

To perform a tenfold cross-validation, the corpus was divided into ten parts. N-fold cross-validation is a method to ensure that results are not based on unrepresentative testing data. Therefore, all experiments are repeated n-times, each time holding out a different n/10th to test the trained model.

4 Extracting evidence

As stated above, the input basically consists of part of speech tags except the words that seemed to bear evidence for the occurrence of a NE. This section describes the extraction of words providing internal and external evidence to the model and thus will be introduced with their word form. All steps described in this section were also performed according to the tenfold cross-validation mentioned above. In keeping with our aims to gain words indicating particular NE categories from our corpus, we developed a simple statistic technique to extract them.

All words occurring around a tag within a window of two words before and two words after a NE were stored first. The chosen frequency threshold *three* led to about 600 tokens. Every entry contained the information about the position and the particular NE category it occurred nearby. Whenever, for instance, the word "Boss" was seen immediately in front of a person's name, it was stored as *1_before_PERSON*. Since there are four window positions and three NE categories, the result is limited to a maximum of twelve classes of indicators.

To reduce the noise in the list, it seemed reasonable to use a TF*IDF weighting. The TF*IDF rates terms by calculating the Term Frequency (TF) in relation to the Inverted Document Frequency (IDF) and is common in information retrieval. The TF*IDF weighting was combined with a measure that describes the probability of occurring at a particular position and a particular NE category. This measure serves to distinguish a word which, for instance, always occurs immediately after a particular NE category from another word which occurs evenly distributed at different positions and near different categories. With a threshold based on this measure, we certainly would not extract words like "CEO", since it is often seen at two positions: Immediately preceding PERSON and two words before ORGANISATION. Therefore, we decided to define a second threshold to extract such words, too. The two thresholds were defined the following way:

- The first requires a word to occur with a probability of 0.7 at the same position and a TF*IDF threshold of 0.5. As an example, a word occurring ten times in the settled window of two words before or after any NE has to occur at least seven times at the same position of a particular NE category. Additionally, its overall frequency must not be higher than 20.
- The second threshold requires a word to be seen with a probability of 0.8 at two positions. This means, for instance, that a word occurring 10 times in the settled window should occur 4 times in front of a person name and 4 times after an organization to fulfil the requirements. Additionally, the same TF*IDF of 0.5 was used.

The extracted list contained about 140 entries on average and seemed to be similar during the ten runs and relatively free of noise.

All words appearing within an organization with a frequency of at least *four* were extracted first. Since our approach avoided any gazetteer-like information, all entries referring to a particular organization had to be removed. Therefore, a list of negative entries was collected, consisting of a list of singletons and a list of tokens occurring together. The former was used to remove entries like "Microsoft" or "AOL", the latter served to remove entries like "Big" and "Blue" or "U.S." and "Robotics". Entries like these were removed whenever the combination of the tokens (in these cases "Big Blue" or "U.S. Robotics") was seen with a frequency higher than 0.8.

The resulting lists, consisting of about 100 elements, showed expected entries like "Ltd.", "GmbH" and "communications", but also some locations like "Hamburg" or "Deutsche". Although they clearly belong to the category LOCATION, they were kept, since their quantity was limited and they were part of an ORGANIZATION and not a LOCATION. A TF*IDF weighting was applied but did not show any effect.

The same was done for the category LOCATION with a frequency threshold of only *two*. As expected, there were almost no results: Only "Bad", a common part of German villages, "New" and "San" were extracted. However, for a larger list and in combination with a compound analysis, this technique would easily extract frequently occurring parts of location names like "St.", "-burg" or "-dorf". The category PERSON was excluded, because it would only generate a list of last names.

5 Preparing the input

Before tagging, the text had to be tokenized. During tokenization, a database table containing abbreviations ending with a dot was consulted to distinguish dots at the end of a sentence from those belonging to a word. Entries referring the magazine like the issue, the article and the author were marked up, so the tagger ignored them.

The part of speech tagging was also done with TnT, using the pre-defined model for German (Brants, 1998) which is trained with the Stuttgart-Tübingen-Tagset (Schiller et al., 1995). The model attempts to differentiate between regular and proper nouns without a further categorization of the latter. Yet, as reported by Müller and Ule (2001), this distinction is error-prone and causes 25% of the part of speech tagging errors. Tagging accuracy of TnT trained on German Text is highly dependent on whether a word was seen during the training or not (86.6% vs. 97.7% accuracy, Brants, 1998). Since our corpus is full of computer vocabulary and thus contains nearly 18% of unknown words on average, a limited tagging accuracy must be assumed.

The output of the part of speech tagger was slightly modified: An additional "*x*" is concatenated to all part of speech tags belonging to words which solely consist of capitals. Usually, they were tagged as noun or proper noun. It seemed reasonable, since that characteristic is evidence for being a company or an organization name. All entries tagged as adjectives, starting with an uppercase letter occurring within and not at the beginning of a sentence are tagged as *ADJAx* for the same reason. The tagger's output after these slight modifications can be seen in the first column of table 2.

Modified Output of the POS tagger		**Baseline**	
...			
<loc>			
Südkoreas	NE	NE	loc_B
</loc>			
größter	ADJA	ADJA	O
Halbleiterhersteller	NN	NN	O
<org>			
Samsung	NN	NN	org_B
Electronics	NE	NE	org_I
</org>			
hat	VAFIN	VAFIN	O
mit	APPR	APPR	O
dem	ART	ART	O
<loc>			
französischen	ADJA	ADJA	loc_B
</loc>			
Halbleiterproduzenten	NN	NN	O
<org>			
SGS	NEx	NEx	org_B
Thomson	NE	NE	org_I
</org>			
...			

Table 2: Modified output of the POS tagger in the first column; the second column shows the same data after tag projection

6 Experiments

To train the recognizer, the tags are projected onto the particular entries. Therefore, the *IOB*-representation was used. *B* denotes the first token of a NE, an *I* any non-initial word of a NE and an *O* is assigned to all tokens not belonging to any NE.

Without integrating the extracted list of words providing internal and external evidence or any further modifications, this is exactly the form of our baseline and can be seen in the second column of table 2. To find out whether the recognizer learns some of the specific patterns of the part of speech tags within and surrounding a NE and whether there were any improvements by integrating our indicators into the corpus, the recognizer was trained on such an input.

For the actual experiment we brought our list of NE indicators into the training corpus. As mentioned above, this list contained entries providing internal as well as entries providing external evidence. The integration of words providing internal evidence was obvious: When replacing the part of speech tag with the corresponding word form, the model learns the frequency of the token and the tags it occurred with. These probabilities generated during the training are stored in the lexicon file. In a randomly chosen lexicon file, for example, the token "corp." occurred six times in the training data and was always tagged as a part of ORGANIZATION. The token "engineering" was seen twelve times and was tagged eight times as ORGANIZATION and four times as not belonging to any NE category.

For the words providing external evidence, a different method was necessary. Since these words usually do not occur within a tag, the same procedure only enabled the recognizer to learn that these words occurred n-times during the training and were tagged n-times with the tag *O*. To provide the model with this external evidence, we had to bring the probabilities of occurring in front or after a particular NE category into the n-gram file. Therefore, additional tags were introduced in two different ways. The input of both experiments can be seen in table 3.

Experiment 1

The recognizer was trained to tag every word providing external evidence uniquely, for which we chose the word form. Apparently, this led to an increase of the tags to be learned, but since these words are always tagged the same way in the training corpus, they are stored in the lexicon file with a probability of 1 to occur with the specific tag. However, of course, the n-gram file had to store every tri-, bi- and uni-gram transition probability of the new tag.

Experiment 2

To reduce the size of the n-gram file, the second experiment attempted to utilize the information delivered from the two different thresholds we described above. The first threshold was designed for words usually seen at the same position of a particular NE category like "Boss" in front of a persons name or "kauft" (engl. "sells") one position after ORGANIZATION. All these words (about 70% of all entries) occurring at the same position of the same category originally tagged with the same part of speech were labeled with a specific tag. Assuming that all these words provide a similar probability we equalized their likelihood of indicating a particular NE. As an example, finite verbs in the list, occurring in front of PERSON, were tagged as *1BPER_VVFIN*. In a randomly chosen lexicon file there are nine verbs occurring with that tag. All of them are verbs of expression like "erklärt" (engl. "explains"), "hofft" ("hopes") etc. In the corresponding n-gram file the tag *1BPER_VVFIN* is listed with an overall occurrence of 48. The model learned (besides other n-grams of the tag) that the next tag was 28 times a NE of the category person, in 19 cases followed by a second person name and nine times by a non tagged token.

All words which crossed the second threshold (i.e. for words not occurring always at the same position but nonetheless seeming to provide reliable evidence) were tagged with their word forms like in experiment 1.

Experiment 3

Just to see how additional information resources influence the results, a last experiment was conducted. During the extraction of the words providing evidence, all words occurring within a NE were stored in the list and therefore introduced with their word form. No further optimization, like an additional filtering of the list or the below described "learn - apply - forget" filter, was performed.

Corresponding words	Input Experiment 1		Input Experiment 2	
...				
Südkoreas	NE	loc_B	NE	loc_B
größter	ADJA	O	ADJA	O
Halbleiterhersteller	NN	O	NN	O
Samsung	NN	org_B	NN	org_B
Electronics	Electr...	org_I	Electr...	org_I
hat	VAFIN	O	VAFIN	O
mit	APPR	O	APPR	O
dem	ART	O	ART	O
französischen	ADJA	loc_B	ADJA	loc_B
Halbleiterproduzenten	Halbleit...	Halbleit...	Halbleit...	1BORG_NN
SGS	Nex	org_B	Nex	org_B
Thomson	NE	org_I	NE	org_I
...				

Table 3: The Input of Experiment 1 and Experiment 2

Table 3 shows a small section of the input of the experiment 1 and the experiment 2. It only differs in the word "Halbleiterproduzenten" which was extracted from training data, since it often precedes an ORGANIZATION. In experiment 1, the model is trained to assign a unique tag identical to the corresponding word form to all these words. In experiment 2, words occurring at the same position of the same category originally tagged with the same part of speech were assigned with a corporate tag. *1BORG_NN* therefore means "one position before ORGANISATION, word tagged as noun".

To increase the likelihood of finding such words, a shallow compound analysis was implemented. Especially since German compound derivation is very productive, it seemed appropriate. The compound analysis handles hyphenated compounds (e.g. "Bull-Gruppe"). Furthermore, it is checked whether the last part of every word which is tagged as a noun and consists of more then seven letters, corresponds to a list entry with at least four letters.

After training a model on nine parts of the annotated corpus, it was applied to the identically pre-processed tenth part of the corpus. This procedure was repeated ten times for each experiment in the tenfold cross-validation manner described above.

Before comparing the output of the recognizer to the manually assigned tags, a "learn - apply - forget" filter, described in Volk and Clematide (2001), was used. It stores all NE found within an article, then the article is "read again" and those NE which have not been found previously are marked up if they have been identified at other positions within the same article. After an article is processed, the system "forgets" the learned NE. This seems adequate, since even human readers sometimes have problems categorizing an isolated token, for example, a token occurring in the header may cause the reader to search for other occurrences of the token.

7 Results

After having applied the above described "learn - apply - forget" filter, the output of the NE recognizer was compared to the manually annotated version and precision and recall were calculated for every category. Precision describes how accurately the recognizer works: It is the number of all the correctly recognized tokens of a particular category divided by the number of all the tokens the recognizer marked as belonging to that category. Recall describes the quantity of the recognizer's result: It is the number of correctly recognized NE tokens divided by the number of NE tokens found in the manually annotated corpus.

	PERSON		ORGANIZATION		LOCATION	
Baseline	P: 42	R: 69	P: 36	R: 35	P: 42	R: 5
Experiment 1	P: 55	R: 90	P: 54	R: 57	P: 69	R: 32
Experiment 2	P: 55	R: 89	P: 54	R: 56	P: 69	R: 32
Experiment 3	P: 71	R: 78	P: 85	R: 62	P: 86	R: 77

Table 4: Results: P = Precision, R = Recall

When comparing the results to other systems, it is apparent that the values are low in general. Yet we should recall that the approach does not use any gazetteer-like information sources and works within the limited context of trigrams. When interpreting the results of the baseline, we can see that it is possible for the recognizer to learn some of the specific patterns of the part of speech tags within and surrounding an ORGANIZATON or a PERSON, albeit on a very low level. However, it seems to be impossible for LOCATION. This corresponds to our expectation, since LOCATIONS are often surrounded by prepositions like "in" or "aus" which are very frequent in German and not specific for LOCATIONS.

Comparing the experiments to the baseline, it is obvious that the utilization of the statistically extracted list produces an effect. The recall for PERSON increases about 20% while precision increases only 10%. Both values for ORGANIZATION increase about 20%, but are still far from being useful. The results of the two experiments differ very slightly, which means that it is appropriate to bundle words with similar features.

Experiment 3 shows the strong effect of using gazetteer-like information, even when it has just been extracted from an annotated corpus. Especially the results for LOCATION were extremely improved. Precision increased for PERSON and ORGANIZATION, while recall did not show much effect for these two categories.

8 Conclusion and further research

Our preliminary experiments show that our approach provides some evidence for the detection and categorization of NE. Both the limited context of a trigram window and the chosen mixture of part of speech tags and statistically extracted words facilitate the recognition of NE. However, when comparing the results to other systems, it is obvious that the proposed approach does not provide enough evidence and an extension of our method is necessary.

When inspecting the recognizer's output, it is apparent that a lot of errors occurred due to the amount of foreign language material within the *Computer Zeitung*. Most of the computer vocabulary and a lot of company names, even German companies, consist of English terms. Even if it was tagged as FM (foreign material) correctly, there is too little information to decide whether it is a NE or not. Yet it is very doubtful if a more accurate part of speech tagger could improve the results.

The second order Markov Model applied on word level, which reduces the context to a trigram window, could simply be too limited for this task. Especially for German, which has a very free word order and therefore, the trigram window often misses the verb (in subordinate clauses positioned at the end), an enhanced approach could be more suitable. The use of the Markov Model on a chunk level and / or other machine learning approaches could be combined with or replace the trigram approach.

The use of gazetteers seems inevitable, especially for the category LOCATION. Any further improvement seems hard to achieve without it, even when they have just been extracted from an annotated corpus.

References

Daniel M. Bikel, Richard Schwartz, Ralph M. Weischedel (1999): An Algorithm that Learns What's in a Name. *Machine Learning (34). Special Issue on Natural Language Learning.* pp. 211-231.

Andrew Borthwick, John Sterling, Eugene Agichtein and Ralph Grishman (1998): Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In: *Proceedings of the Sixth Workshop on Very Large Corpora. Montreal, Canada*.

Thorsten Brants (1998): TnT - A Statistical Part-of-Speech Tagger. Saarland University, Computational Linguistics. Saarbrücken. Online available: http://www.coli.uni-sb.de/~thorsten/tnt/.

Thorsten Brants (2000): TnT - a statistical part-of-speech tagger. In: *Proceedings of the 6th Applied NLP Conference, ANLP-2000, April 29 -- May 3, 2000, Seattle, WA.*

CZ (1998): Konradin-Verlag. Computer Zeitung auf CD-ROM. Volltextrecherche aller Artikel der Jahrgänge 1993 bis 1996. Konradin Verlag. Leinfelden-Echterdingen.

David D. McDonald (1996): Internal and External Evidence in the Identification and Semantic Categorization of Proper Names. In: Bran Boguraev and James Pustejovsky (editors): *Corpus Processing for Lexical Acquisition*. pp. 21-39. MIT Press. Cambridge, MA.

Andrei Mikheev, Claire Groover and Marc Moens (1998): Description of the LTG System Used for MUC-7. In: *Proceedings of MUC-7. Fairfax, Virginia. 1998.*

Andrei Mikheev, Marc Moens and Claire Grover (1999): Named Entity recognition without gazetteers. In: *EACL'99, Bergen, Norway. June 1999*. pp. 1-8

Frank H. Müller and Tylman Ule (2001): Satzklammer annotieren und Tags korrigieren. Ein mehrstufiges 'Top-Down-Bottom-Up'-System zur flachen, robusten Annotierung von Sätzen im Deutschen. In: Henning Lobin (editor): *Sprach- und Texttechnologie in digitalen Medien. Proceedings der GLDV-Frühjahrstagung 2001*. Norderstedt. Book on Demand. pp. 225-234.

Günther Neumann and Jakub Piskorski (2000): An intelligent text extraction and navigation system. In: *6th International Conference on Computer-Assisted Information Retrieval (RIAO-2000). Paris.*

Anne Schiller, Simone Teufel and Christine Thielen (1995): Ein kleines und erweitertes Tagset fürs Deutsche. In: Helmut Feldweg, Erhard W. Hinrichs (editors): *Wiederverwendbare Methoden und Ressourcen zur linguistischen Erschließung des Deutschen. Volume 73 of Lexicographica Series Maior.* Niemeyer Verlag. Tübingen. pp. 193-203.

Martin Volk and Simon Clematide (2001): Learn - Filter - Apply - Forget. Mixed Approaches to Named Entity Recognition. In: *Proceedings of 6th International Workshop on Applications of Natural Language for Information Systems. Madrid 2001.*

GuoDong Zhou, Jian Su (2002): Named Entity Recognition using an HMM-based Chunk Tagger. To be published in: *Proceedings of the 40th Conference of Association for Computational Linguistics (ACL-02), University of Pennsylvania, Philadelphia 2002.*