Adapting Data Mining for German Named Entity Recognition

Damien Nouvel, Jean-Yves Antoine

Adapting Data Mining for German Named Entity Recognition

Damien Nouvel
Université François Rabelais Tours
Laboratoire d'informatique
Tours, France
damien.nouvel@limsi.fr

Jean-Yves Antoine
Université François Rabelais Tours
Laboratoire d'informatique
Tours, France
jean-yves.antoine@univ-tours.fr

Abstract

In the latest decades, machine learning approaches have been intensively experimented for natural language processing. Most of the time, systems rely on using statistics within the system, by analyzing texts at the token level and, for labelling tasks, categorizing each among possible classes. One may notice that previous symbolic approaches (e.g. transducers) where designed to delimit pieces of text. Our research team developped mXS, a system that aims at combining both approaches. It locates boundaries of entities by using sequential pattern mining and machine learning. This system, intially developped for French, has been adapted to German.

1 Introduction

In the 90's and until now, several symbolic systems have been designed that make intensive use of regular expressions formalism to describe Named Entities (NEs). Those systems combine external and internal evidences (McDonald, 1996), as patterns describing contextual clues and lists of names per NE category. Those systems achieve high accuracy for NE Recognition (NER), but, because they depend on the handcrafted definition of lexical ressources and detection rules, their coverage remains an issue.

To address NER, machine learning usually states the problem as categorizing words that belong to a NE, taking into account various clues (features) in a model that is automatically parametrized by leveraging statistics from a training corpus. Among these methods, some only focus on the current word under examination (maximum entropy, SVM) (Borthwick et al., 1998), while others also evaluate stochastic dependencies (HMM, CRF) (McCallum and Li, 2003; Ratinov and Roth, 2009). Most of the time, those approaches output the most probable sequence of labels for a given sentence. This is generally known as the "labeling problem", applied to NER.

Many approaches rely on pre-processing steps that provide additional information about data, often Part-Of-speech (POS) tagging and proper names lists, to determine how to automatically tag texts (Ratinov and Roth, 2009). Recently, data mining techniques (Freitag and Kushmerick, 2000) have been experimented, but we are not aware of work that goes beyond the step of extracting patterns for NER.

Our system, mXS[1] (Nouvel et al., 2014), automatically mines patterns and use them as features for machine learning. It focuses on boundaries of NEs, as beginning or ending tags to be inserted. Internally, the system considers each tag delimiting a NE as an item of interest and extracts detection rules (which may be used as feature but also may be read by humans). To the best of our knowledge, this way of combining symbolic and machine learning approaches is original in the framework of NER. It obtained satisfying results

[1] https://github.com/eldams/mXS

during the ANR ETAPE of the ANR French research agency evaluation campaign, ranked 3rd or 2nd among 10 particpants. This paper presents our adaptation of mXS to German.

2 Coding, Preprocessings and Lexicon

2.1 Coding NEs beyond BIO Format

As previously mentioned, most of the approaches for doing NER rely on labelling tokens of a text. This leads to representations as illustrated in Figure 1 where each token is assigned a dedicated class. Machine learning approaches are known to be efficient to solve this kind of problem. Our main concern about this representation is that it is now mandatory to classify all tokens within a named entity, even underspecific tokens such as `für/I-ORG`.

As a result, mXS uses internally a different coding to represent NE tokens: only beginning and ending of NEs are explicitly mentionned, in a XML-like fashion, e.g. `<PER> Cartier </PER>`. Our goal is then to discover the correct positions where NE tags have to be inserted, as showed in Figure 2. This approach doesn't prevent to use machine learning techniques, avoids the artificial split of NE classes (e.g. `B-XXX` and `I-XXX`) and can be used in combination with sequential data mining techniques.

2.2 Morphosyntax

Initial preprocessings and linguistic analysis are done using TreeTagger (Schmid, 1994), that conjointly tokenizes, lemmatizes and assigns POS to each token. Our first experiments demonstrate that this software gives sufficient clues, especially by identifying proper names, to ground our system. We use this information, as gradual generalizations for building representation of texts. Consider for instance this sentence from the GermEval training corpus:

```
Der <LOC> Queen <PER> Sirikit </PER>
Park </LOC> ist ein Botanischer Garten
```

Here, Botanischer is progressively generalized as `botanisch` (lemma) then `ADJA` (adjective POS). This incremental generalization is described by `ADJA/botanisch/Botanischer` where the `/` symbol is used as a specialization operator. Our text mining process is able to consider for any token all possible generalizations over this hierarchy[2]. The sentence is now represented as:

```
ART/die/Der <LOC> NN/Queen/Queen
<PER> NN/Sirikit/Sirikit </PER>
NN/Park/Park </LOC> VAFIN/sein/ist
ART/eine/ein ADJA/botanisch/Botanischer
NN/Garten/Garten
```

As data mining process is aimed at extracting generic patterns, we exclude surface variations (but keep their lemmas) and lexicalization of proper names (to avoid overfitting) when preprocessing training corpus:

```
ART/die <LOC> NN/Queen <PER> NN/Sirikit
</PER> NN/Park </LOC> VAFIN/sein
ART/eine ADJA/botanisch NN/Garten
```

The French version of mXS includes many dedicated adaptations to improve recognition of specific linguistic expressions. The German version of mXS that participates to GermEval does not include such useful improvements.

2.3 Lexicon

In the experiments presented in Section 4, the baseline system does not use any lexicon, and thus only relies on morphosyntax analysis. To improve performance, we also considered three proper noun lexicons as additional resources (Table 1): ST is extracted from FreeBase ; IP and IW are gross-grained and fine-grained versions of a lexicon extracted from Wikipedia (Savary et al., 2013). They implement usual classes for NER as antrhroponyms, toponyms, first names, last names, organizations, etc.

Lexicon	Categories	Entries
ST	5	497 093
IP	7	33 167
IW	118	33 167

Table 1: System lexicons number of classes and entries

Those lexicons provide another possible level of generalization. As it is more related to semantic properties of tokens, this information will be considered as the top level to generalize tokens. mXS also supports multiword expressions and ambiguity at any level: semantic categories

[2]Besides, as it is not a column format, the number of possible generalizations may vary from one token to another

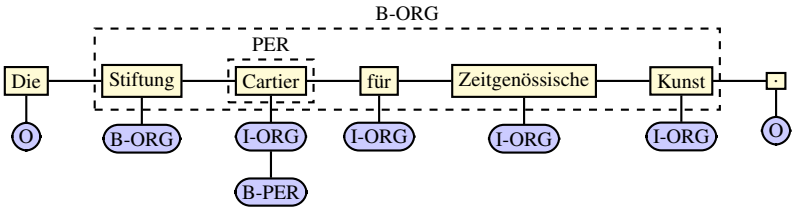

Figure 1: Annotation as a labelling task

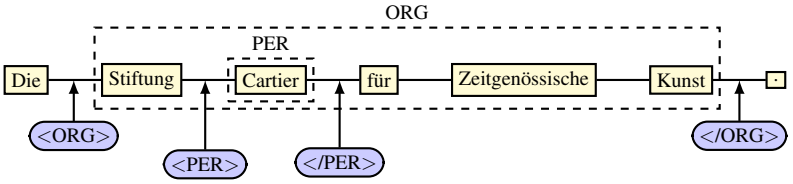

Figure 2: Annotation as an annotation task

provided by lexicons may be assigned to multiple tokens, and each token may receive multiple categories. Using those lexicons adds information:

```
  -/ART/die/Der <LOC>
Organizations/NN/Queen/Queen
<PER> -/NN/Sirikit/Sirikit
</PER> -/NN/Park/Park </LOC>
-/VAFIN/sein/ist -/ART/eine/ein
Locations/ADJA/botanisch/Botanischer
Locations/NN/Garten/Garten
```

Furthermore, for TreeTagger categories `NN` and `NE`, suffixes with a size of 3 or 4 characters are also considered as an intermediate generalization level, e.g. `Locations/NN/Garten` now becomes `Locations/NN/SUFF:ten/SUFF:rten/Garten`. This also illustrates how hierarchical sequential mining can easily fit special needs (e.g. language or task adaptation of preprocessings).

3 Sequential Data Mining to extract Patterns as Features

Mining techniques are applied on the information provided by preprocessings. The data miner within mXS proceeds in a supervised level-wise fashion to extract generalized sequential patterns (Agrawal and Srikant, 1995) that are correlated to NE tags. To limit complexity, the search is limited by criterions such as minimum support (frequency), minimum confidence (regarding the presence of NE tags) and redundancy within patterns. Extracted patterns are supposed to be valuable clues for detecting NE boundaries. Due to a lack of space, the mining process will not be detailed in this paper, further information can be found in (Nouvel et al., 2014).

mXS implements hierarchical mining: patterns are sequences of diversely generalized natural language tokens and enriched data and NE tags. Here are some examples of extracted patterns:

```
<PER> NE ART NN/SUFF:ung
<LOC> CITY/NN APPR/in REGION/NE </LOC>
<PER> NE NN APPR CITY </LOC>
```

The extracted patterns are used as features by a maxent classifier, provided by the scikit-learn toolkit (Pedregosa et al., 2011) that estimates, at any position of a sentence, the probability to insert tags given the patterns. using a Viterbi algorithm, the decoding step combines individual probabilities to select annotation that maximizes likelihood. The advantage of this approach, besides avoiding the artificial split of B- and I- of BIO format, is that it can insert multiple tags at a given position, enabling recursive annotation as required by the GermEval campaign.

4 Experiments and Results

We assess the usefulness of the extracted patterns for NER, by selecting them at different thresholds of support and confidence. Table 2 shows that best score are obtained with low support (5) and medium confidence (10%). Around 17000

patterns are extracted with these parameters. The comparison with situations where pattern features are not used ("inf") shows that patterns always lead to better performances, reaching a maximum increase of +2.5% of the overall f-score.

supp	conf%	rules	fscore%	prec%	rec%
5	5	21 620	59.50	76.44	48.71
5	10	17 268	**59.91**	76.76	**49.13**
5	50	7 512	58.87	76.87	47.70
10	5	9 505	59.62	76.82	48.71
10	10	7 460	59.55	76.68	48.67
10	50	3 108	58.53	76.80	47.28
50	5	1 283	59.41	77.37	48.22
50	10	972	59.35	**77.42**	48.11
50	50	359	58.35	77.03	46.96
inf	inf	0	57.41	76.01	46.12

Table 2: Score without lexicon

We investigated the benefits of using three lexicons, separately or jointly. As displayed in Table 3, using them always lead to significant improvement. Unfortunately, combining them degrades performances (we assume that those resources are not as complementary as expected).

lex	supp	conf%	fscore%	prec%	rec%
none	5	10	59.91	76.76	49.13
ST	50	50	**62.97**	**80.63**	**51.66**
IP	10	10	61.07	78.83	49.84
IW	5	20	60.38	78.10	49.22
All	50	10	62.71	80.61	51.31

Table 3: Score depending on lexicon

We built our final system using only the ST lexicon, which provided the best score (63.16), each run being a combination of frequency and confidence parameters. Official results in Table 4 are close to what has been obtained on the development dataset and unfortunately confirmed our very high precision but unsufficient recall: our system is ranked 7th out of 11. We suspect overfitting and conducted additional experiments for fine-tuning maxent regularization parameter. For the moment, this leads to a better f-score (64.19) over the official test data, without clarifying the question of the strong difference between precision (80.76) and recall (53.26).

supp	conf%	fscore%	prec%	rec%
5	10	61.63	79.05	50.5
10	50	62.29	80.46	50.81
50	50	**62.39**	**80.62**	**50.89**

Table 4: Final scores

5 Conclusion

This paper shows how to use data mining in an original way (separate detection of NE boundaries instead of BIO tagging) to implement a rather efficient multilevel named entity recognition system. Adapting mXS from French to German was quite easy, thanks to the availability of resources. Obviously, this version of mXS lacks linguistic adaptations specific to German, what prevent us to reach an optimal level of performance. Nevertheless, we reached our main goal, which was to assess the reliability of our original approach on another language using similar preprocessings steps and our generic pattern mining implementation.

Acknowledgments

Thanks to people from LIMSI-CNRS and ANR ETAPE for endorsing our work on NER.

References

Rakesh Agrawal and Ramakrishnan Srikant. 1995. Mining sequential patterns. In *ICDE*, pages 3–14.

Andrew Borthwick, John Sterling, et al. 1998. Exploiting diverse knowledge sources via maximum entropy in NER. In *WVLC'1998*.

Dianne Freitag and Nicholas Kushmerick. 2000. Boosted wrapper induction. In *WMLIE*.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *CONLL'2003*.

David D. McDonald. 1996. Internal and external evidence in the identification and semantic categorization of proper names. *CPLA*, pages 21–39.

Damien Nouvel, Jean-Yves Antoine, and Nathalie Friburger. 2014. Pattern mining for named entity recognition. *LNCS/LNAI Series*, 8387i.

Fabian Pedregosa, Gaël Varoquaux, et al. 2011. Scikit-learn: Machine learning in Python. *J. of Machine Learning Research*, 12:2825–2830.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in NER. In *CONLL'2009*, pages 147–155, Stroudsburg, PA, USA. ACL.

Agata Savary, Leszek Manicki, and Malgorzata Baron. 2013. Populating a multilingual ontology of proper names from open sources. *J. Language Modelling*, 1(2):189–225.

Helmut Schmid. 1994. Probabilistic POS tagging using decision trees. In *NMLP*.