Learning Categories and their Instances by Contextual Features

Antje Schlaf, Robert Remus

Natural Language Processing Group, University of Leipzig, Germany
{antje.schlaf, rremus}@informatik.uni-leipzig.de

Abstract

We present a 3-step framework that learns categories and their instances from natural language text based on given training examples. Step 1 extracts contexts of training examples as rules describing this category from text, considering part of speech, capitalization and category membership as features. Step 2 selects high quality rules using two consequent filters. The first filter is based on the number of rule occurrences, the second filter takes two non-independent characteristics into account: a rule's precision and the amount of instances it acquires. Our framework adapts the filter's threshold values to the respective category and the textual genre by automatically evaluating rule sets resulting from different filter settings and selecting the best performing rule set accordingly. Step 3 then identifies new instances of a category using the filtered rules applied within a previously proposed algorithm. We inspect the rule filters' impact on rule set quality and evaluate our framework by learning first names, last names, professions and cities from a hitherto unexplored textual genre – search engine result snippets – and achieve high precision on average.

Keywords: Named Entity Recognition, Information Extraction, Text Mining

1. Introduction

A crucial aspect of text understanding is the knowledge of certain categories and their instances, e.g. knowing that "teacher", "engineer" and "baker" are instances of the category "profession". Exhaustive knowledge of this kind is particularly essential in environments like task-specific search engines or information extraction systems (Appelt, 1999). In this paper, we present a fully automatic 3-step framework that learns categories and their instances from natural language text based on given training examples. Our approach is based on the assumption, that instances of the same category share similar contexts. We only use example instances of a category and text to learn from, as this setup reflects a real world scenario of identifying category instances. We evaluate the framework's performance by learning instances of 4 categories from search engine result snippets obtained from the people search engine `yasni.de`.

Although the framework we present is not necessarily limited to learning named entities, its main purpose is to do so. Thus, it can be seen as an instance of named entity recognition (NER). NER has been widely studied since the mid-90s: Nadeau and Sekine (2007) provide a comprehensive survey of the field. NER was approached through supervised (McCallum and Li, 2003), semi-supervised, and unsupervised learning methods (Etzioni et al., 2005). Closest to our work is the algorithm proposed by Riloff and Jones (1999). Wang et al. (2009) also learn semantic classes for query understanding.

This paper is structured as follows: In the next Section we present our framework. In Section 3. we describe our experimental setup and evaluate its results. Finally, we draw conclusions and point out possible directions for future research in Section 4..

2. Learning Categories and their Instances

As shown in Figure 1, we first learn rules for a category by extracting contexts of initial category instances from their occurrences in text. Then we automatically select threshold values for 2 consequent rule filters to determine threshold values particularly adapted to the category and the underlying textual genre. Finally, we apply the filtered rules to learn new instances within a previously proposed algorithm, Biemann (2003)'s *pendulum*.

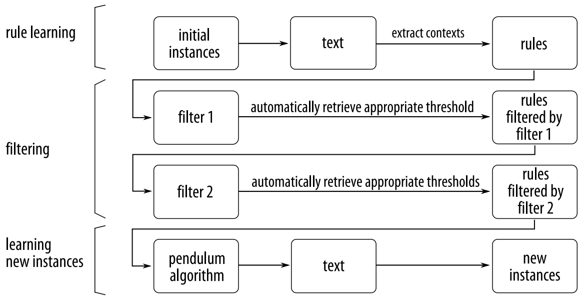

rule learning
initial instances
text
extract contexts
rules
filtering
filter 1
automatically retrieve appropriate threshold
rules filtered by filter 1
filter 2
automatically retrieve appropriate thresholds
rules filtered by filter 2
learning new instances
pendulum algorithm
text
new instances

Figure 1: Framework overview.

Furthermore, by inspecting the automatic evaluation results of rules filtered by different threshold values we try to estimate the effects different rule filters have on the overall rule set quality.

2.1. Learning Rules

Starting with initially known instances of a category, we retrieve their occurrences in given text and learn rules by extracting feature values from the instance itself and terms around it. These rules can later be applied to text to learn instances of a certain category.

In general any word-based feature may be used. Contextual features considered in the evaluation of our framework are *capitalization*, *part of speech* (POS) and *category membership*, i.e. whether a term is an instance of a certain category or not. Figure 2 shows an exemplary rule that "learns" first names. Columns represent terms with the instance being the term in the middle, rows represent required feature values of capitalization, category membership and POS.

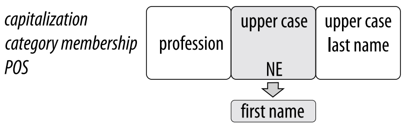


Figure 2: An exemplary rule that learns first names.

2.2. Automatic Rule Evaluation

The quality of a rule or whole rule set is assessed by measuring their precision, recall and f-score when applying the respective rule(s) to an evaluation text. Evaluating learned instances as well as retrieving all relevant instances from text requires huge manual effort. Instead, we automatically evaluate the rules by just referring to already known instances, i.e. the instances we initially learned the rules with.

This automatic evaluation is only an approximation and may differ greatly from a manual evaluation. Only initially known instances can be classified as correct or retrieved as relevant from the evaluation text. Therefore, the automatically calculated precision describes a lower bound of the real precision, because the unknown and therefore as false classified instances may actually contain true instances. Apart from that, we do not aim for an automatically calculated precision of 1.0, as this would imply that no new instances were acquired by our rules.

The instances learned by the final filtered rule set are then evaluated manually to determine their actual precision. A manual recall and f-score evaluation was not performed due to the huge manual effort of retrieving relevant instances from the evaluation text. From now on, all mentioned measures are automatically determined, if not stated otherwise.

2.3. Filtering Rules

After learning rule sets, we improve their quality by two consequent rule filters. Filter 1 selects rules that were extracted multiple times and therefore exceed a certain *threshold occurrence*. The underlying rationale is that rules with a more frequent occurrence are likely to be more reliable than low-frequency rules. Filter 2 takes two non-independent characteristics into account. The first being whether rules reproduce known instances with a certain minimal ratio and thus, fulfill a certain threshold precision. The second being whether rules are "productive" or not, i.e. if they extract a certain amount of instances, regardless if they are known to be correct or not: the threshold number of learned instances.

We inspect various threshold values per filter separately for all categories (cf. Figure 3 and 4). We apply each rule set resulting from filtering with certain threshold settings to the evaluation text and automatically evaluate its learned instances as described in the previous section. This is done for two reasons: First, we inspect the impact of thresholds on the consequent rule set. Secondly, we automatically select an appropriate threshold value which adapts to the particular category and the textual genre. We optimize threshold values for maximal harmonic mean of precision P and recall R, which equals f-score F, as well as the maximal harmonic mean H of P, R, and the number of learned instances $\#L$. This optimization of threshold values based on automatic evaluation is only an approximation of the actual optimal values, but still it requires no manual effort.

2.4. Learning Instances

The resulting rule sets are then used for their actual purpose: learning new instances. To learn new instances it is possible to just apply the rule sets to text. However, we learn instances by utilizing an algorithm, known for both its high precision and recall in identifying named entities and relations from natural language text: Biemann (2003)'s pendulum. We slightly modified pendulum for our own purposes: We perform candidate identification and verification on a fixed amount of text, and we skip its proposed iterative learning.

3. Experiments

We now describe the experiments carried out to evaluate the framework's quality. We obtained German-language data, so called search engine *result snippets*, from the people search engine `yasni.de`. Result snippets are a hitherto *unexplored textual genre* and typically look like

> Max Mustermann, Elektroinstallateure #TITLE#
> in Musterstadt, Musterstr. 8, Tel.: (0123) 45678

or

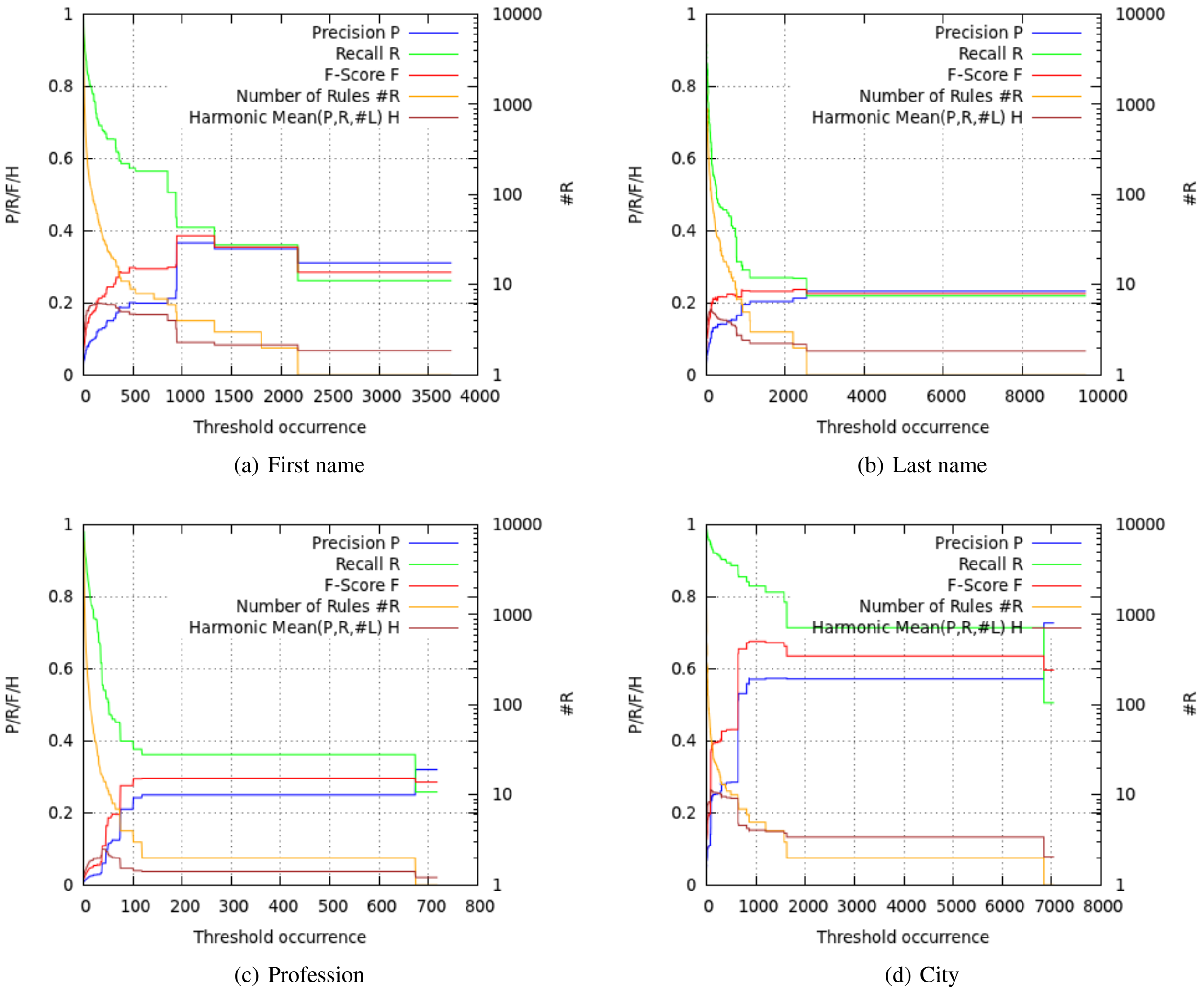
> Ich bin dann mal alt! – Johannes Pausch; Gert Böhm | neues Buch ...#TITLE# Johannes Pausch; Gert Böhm – Ich bin dann mal alt! – Dem Leben auf der Spur ...

In general our framework will work with any text type by adapting to it through automatically learned rules and filter thresholds. For our initial experiments, we decided to use such "dense" data because it is highly likely to contain plenty interesting categories and their instances. Our corpus used for learning rules consists of roughly 10 million result snippets. The automatic rule evaluation was performed on a randomly selected subset of 100,000 result snippets. Additionally, to retrieve the rule sets' actual precision, rule sets that lead to the best automatically retrieved evaluation results were manually evaluated by human annotators.

3.1. Results

3.1.1. Learning Rules

We learned rules for 4 categories: first name, last name, city and profession. The according number of initially known instances were: first name (13,496), last name (17,148), city (6,843), and profession (2,411). For each instance of a certain category we extracted word-based feature values from a maximum of 10 randomly selected result snippets containing that particular instance. As mentioned earlier, considered features are capitalization, POS tags obtained by Stanford POS tagger (Toutanova and Manning, 2000; Toutanova et al., 2003) and category membership. A window of size 5 was used, i.e. the instance plus 2 words before and 2 words after it. Table 1 shows the results of the rule learning.



(a) First name

(b) Last name

(c) Profession

(d) City

Figure 3: Filter 1's impact on rule sets learned for first names, last names, professions and cities.

3.1.2. Filtering Rules by Threshold Occurrence

Figure 3 shows the impact of various threshold values on the consequent rule set. As rule occurrence is roughly *Zipf-like distributed* (Zipf, 1972), even a low threshold occurrence leads to a huge reduction of the rule set. As the threshold occurrence is a simple pre-filter and the main interest lies in evaluating filter 2, reduction of the rule set by filter 1 should not be too strong. Since automatic selection of the threshold value based on maximum F leads to less than 10 rules per category, the selection was based on maximum H. The respective results of the automatic and the manual evaluation are shown in Table 1. The reduction of the rule set is still very strong for all categories, while F and H constantly increase.

3.1.3. Filtering Rules by Threshold Precision and Threshold Number of Learned Instances

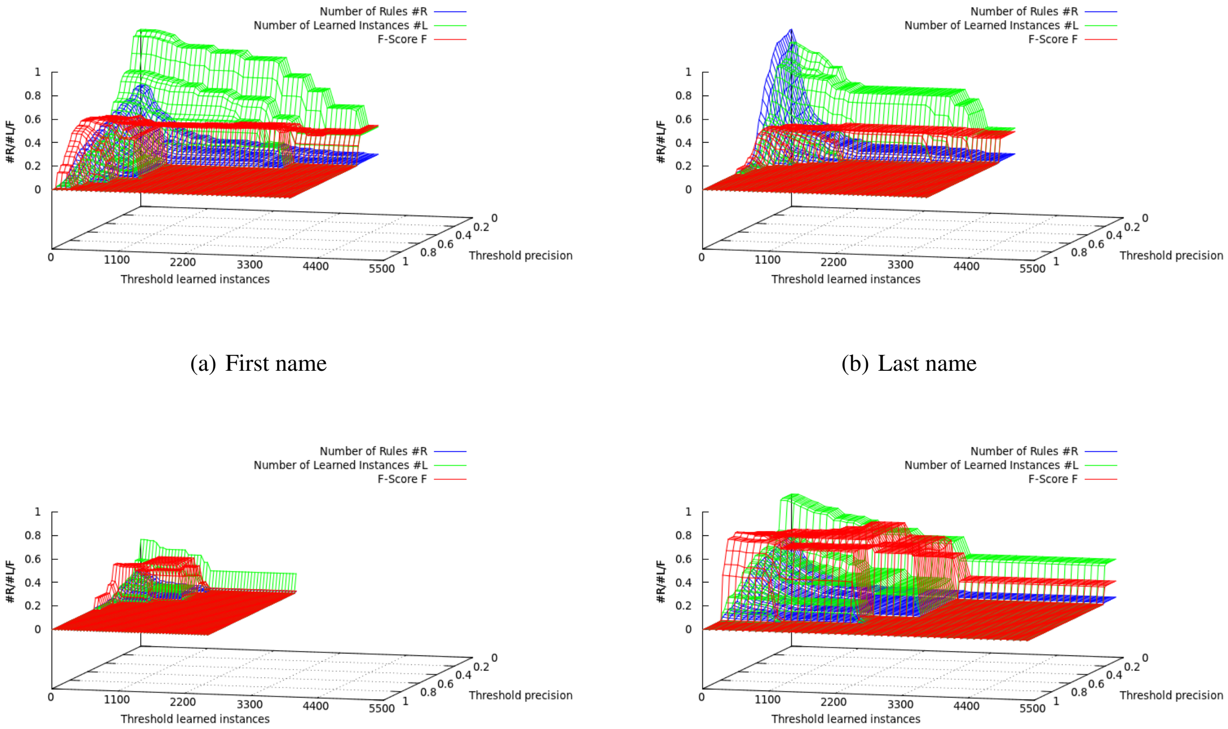
For filter 2 we build upon the previously filtered rule set and perform a grid search on the threshold values of both threshold precision and threshold of learned instances to simultaneously optimize their impact on the rule set. For brevity we only plot the number of rules, number of learned instances and F in Figure 4.

Though the figures of all 4 categories look different, they all allow the following conclusion: The rule set size can be reduced drastically by selecting a small value above zero for threshold number of learned instances without losing much in learned instances and F.

The optimization of both threshold values is based on maximum F. Thresholds selected based on maximum H were calculated as well, but lead to worse results and are therefore not presented here. The respective results of the automatic and manual evaluation are presented in Table 1. Whereas first name and profession only reach medium quality in manually determined precision (0.664 and 0.648), last name and city reach very high quality (0.958 and 0.996).

3.1.4. Learning Instances

Finally, the filtered rule set is used to learn *new* instances using pendulum; hence, instances already known are not considered in this step. The results are shown in Table 2. Again, first name and profession reach medium quality while last name and city reach very high quality in precision.



(a) First name

(b) Last name

(c) Profession

(d) City

Figure 4: Filter 2's impact on rule sets learned for first names, last names, professions and cities.

Category	State	Rules	Instances	Precision	Manual Precision	Recall	F-Score	H
First name	Original	24,690	89,592	0.032	–	0.986	0.061	0.089
	after Filter 1	53	16,213	0.124	0.193	0.704	0.211	0.2
	after Filter 2	19	2,566	0.496	0.664	0.445	0.469	0.077
Last name	Original	22,910	92,769	0.029	–	0.986	0.056	0.083
	after Filter 1	100	14,461	0.118	0.756	0.623	0.198	0.184
	after Filter 2	62	4,834	0.211	0.958	0.373	0.269	0.116
Profession	Original	6,520	76,065	0.009	–	0.98	0.018	0.027
	after Filter 1	16	6,549	0.06	0.154	0.555	0.198	0.093
	after Filter 2	4	719	0.299	0.648	0.305	0.302	0.023
City	Original	5,605	75,744	0.041	–	0.998	0.079	0.113
	after Filter 1	43	12,804	0.23	0.538	0.944	0.369	0.242
	after Filter 2	10	3,201	0.68	0.996	0.699	0.69	0.097

Table 1: Results of "learning rules".

3.2. Discussion

Wrongly learned first names include professions, last names or business and location descriptions. Wrongly learned professions often describe profession branches or certain details associated with them, e.g. "consulting" or "design", instead of being actual professions. Since those terms are widely used to describe professions, the category profession may be loosened to *profession description*. Consequently, a higher precision would be reached.

We note, manually evaluated precision differs noticeably from automatically calculated precision across all categories. Furthermore, whereas inspecting the automatically calculated precisions of first names and professions might lead to the conclusion that the manually evaluated precision of first names is also higher than the professions', the actual manual evaluation states quite the opposite. Hence, we cannot directly infer the real precision value from its "approximation" to estimate the actual quality of a rule set or to compare evaluation results of different categories. Nevertheless, automatically evaluating rule sets allows us to automatically select category and text type adapted threshold values for filters that improve the overall rule set quality without any manual effort.

Category	Instances	Manual Precision
First name	441	0.667
Last name	2,644	0.977
Profession	231	0.751
City	929	1.0
Average	1,056.75	0.849

Table 2: Results of “learning instances”.

4. Conclusions & Future Work

We proposed a 3-step framework for learning categories and their instances and deeply investigated the effects 2 rule filters have. We achieved an average precision of 0.849 for learning instances of 4 categories: first name, last name, profession and city.

Future research directions include learning more categories, learning from text types different from the one used in this work, such as newspaper articles or blog posts, and learning based on other features, such as affixes and sequence positioning. We would also like to investigate (Biemann, 2003)’s proposed iterative learning and evaluate other rule filter criteria.

5. Acknowledgements

This research was funded by Sächsische AufbauBank (SAB) and European Regional Development Fund (EFRE). We gratefully acknowledge the effort of our annotators at `yasni.de` and thank them for providing us data and insights into their work.

6. References

D.E. Appelt. 1999. Introduction to Information Extraction. *AI Communications*, 12(3):161–172.

C. Biemann. 2003. Extraktion von semantischen Relationen aus natrlichsprachlichem Text mit Hilfe von maschinellem Lernen. In *Sprachtechnologie fr multilinguale Kommunikation, Beitrge der GLDV-Frhjahrstagung*.

O. Etzioni, M. Cafarella, D. Downey, A.M. Popescu, T. Shaked, S. Soderland, D.S. Weld, and A. Yates. 2005. Unsupervised Named-entity Extraction from the Web: An Experimental Study. *Artificial Intelligence*, 165(1):91–134.

A. McCallum and W. Li. 2003. Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-enhanced Lexicons. In *Proceedings of the 7th Conference on Natural language learning (CoNLL)*, pages 188–191.

D. Nadeau and S. Sekine. 2007. A Survey of Named Entity Recognition and Classification. *Lingvisticae Investigationes*, 30(1):3–26.

E. Riloff and R. Jones. 1999. Learning Dictionaries for Information Extraction by Multi-level Bootstrapping. In *Proceedings of the National Conference on Artificial Intelligence*, pages 474–479.

K. Toutanova and C.D. Manning. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-speech Tagger. In *Proceedings of Joint Conference on Empirical Methods in Natural Language Processing (EMNLP) and Very Large Corpora (VLC)*, pages 63–71.

K. Toutanova, D. Klein, C.D. Manning, and Y. Singer. 2003. Feature-rich Part-of-speech Tagging with a Cyclic Dependency Network. In *Proceedings of the Human Language Technologies: North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 173–180.

Y.Y. Wang, R. Hoffmann, X. Li, and J. Szymanski. 2009. Semi-supervised Learning of Semantic Classes for Query Understanding: from the Web and for the Web. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, pages 37–46.

G.K. Zipf. 1972. *Human Behavior and the Principle of Least Effort*. Hafner, New York.