Self-Annotation for Fine-Grained Geospatial Relation Extraction

Andre Blessing **Hinrich Schütze**
Institute for Natural Language Processing
Universität Stuttgart
ner@ifnlp.org

Abstract

A great deal of information on the Web is represented in both textual and structured form. The structured form is machine-readable and can be used to augment the textual data. We call this augmentation – the annotation of texts with relations that are included in the structured data – *self-annotation*. In this paper, we introduce self-annotation as a new supervised learning approach for developing and implementing a system that extracts fine-grained relations between entities. The main benefit of self-annotation is that it does not require manual labeling. The input of the learned model is a representation of the free text, its output structured relations. Thus, the model, once learned, can be applied to any arbitrary free text. We describe the challenges for the self-annotation process and give results for a sample relation extraction system. To deal with the challenge of fine-grained relations, we implement and evaluate both shallow and deep linguistic analysis, focusing on German.

1 Introduction

In the last years, information extraction has become more important in domains like context-aware systems (e.g. Nexus (Dürr et al., 2004)) that need a rich knowledge base to make the right decisions in different user contexts. Geospatial data are one of the key features in such systems and need to be represented on different levels of detail. Data providers do not cover all these levels completely. To overcome this problem, *fine-grained* information extraction (IE) methods can be used to acquire the missing knowledge. We define fine-grained IE as methods that recognize entities at a finer grain than standard categories like person, location, and organization. Furthermore, the quality of the data in context-aware systems plays an important role and updates by an information extraction component can increase the overall user acceptance.

For both issues an information extraction system is required that can handle *fine-grained relations*, e.g., "X is a suburb of Y" or "the river X is a tributary of Y" – as opposed to simple containment. The World Wide Web offers a wealth of information about geospatial data and can be used as source for the extraction task. The extraction component can be seen as a kind of sensor that we call *text senor* (Blessing et al., 2006).

In this paper, we address the problem of developing a flexible system for the acquisition of relations between entities that meets the above desiderata. We concentrate on *geospatial* entities on a fine-grained level although the approach is in principle applicable to any domain. We use a supervised machine learning approach, including several features on different linguistic levels, to build our system. Such a system highly depends on the quality and amount of labeled data in the training phase. The main contribution of this paper is the introduction of self-annotation, a novel approach that allows us to eliminate manual labeling (although training set creation also involves costs other than labeling). Self-annotation is based on the fact that Word Wide Web sites like Wikipedia include, in addi-

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 80–88,
Beijing, August 2010

tion to unstructured text, structured data. We use structured data sources to automatically annotate unstructured texts. In this paper, we use German Wikipedia data because it is a good source for the information required for our context-aware system and show that a system created without manual labeling has good performance.

Our trained model only uses text, not the structured data (or any other markup) of the input documents. This means that we can train an information extractor on Wikipedia and then apply it to any text, regardless of whether this text also contains structured information.

In the first part of this paper, we discuss the challenges of self-annotation including some heuristics which can easily be adapted to different relation types. We then describe the architecture of the extraction system. The components we develop are based on the UIMA (Unstructured Information Management Architecture) framework (Hahn et al., 2008) and include two linguistic engines (OpenNLP[1], FSPar). The extraction task is performed by a supervised classifier; this classifier is also implemented as a UIMA component and uses the ClearTK framework. We evaluate our approach on two types of fine-grained relations.

2 Related work

Jiang (2009) also addresses the issue of supervised relation extraction when no large manually labeled data set is available. They use only a few seed instances of the target relation type to train a supervised relation extraction system. However, they use multi-task transfer learning including a large amount of labeled instances of other relation types for training their system. In contrast, our work eliminates manual labeling by using structured data to annotate the relations.

Wu and Weld (2007) extract facts from infoboxes and link them with their corresponding representation in the text. They discuss several issues that occur when using infoboxes as a knowledge base, in particular, (i) the fact that infoboxes are incomplete; and (ii) *schema drift*. Schema drift occurs when authors over time use different attribute names to model facts or the same attributes are used to model different facts. So the semantics of the infoboxes changes slightly and introduces noise into the structured information. Their work differs from self-annotation in that they are not interested in the creation of self-annotated corpora that can be used as training data for other tasks. Their goal is to develop methods that make infoboxes more consistent.

Zhang and Iria (2009) use a novel entity extraction method to automatically generate gazetteers from seed lists using Wikipedia as knowledge source. In contrast to our work they need structured data for the extraction while our system focuses on the extraction of information from unstructured text. Methods that are applicable to any unstructured text (not just the text in the Wikipedia) are needed to increase coverage beyond the limited number of instances covered in Wikipedia.

Nothman et al. (2009) also annotate Wikipedia's unstructured text using structured data. The type of structured data they use is hyperlinking (as opposed to infoboxes) and they use it to derive a labeled named entity corpus. They show that the quality of the annotation is comparable to other manually labeled named entity recognition gold standards. We interpret their results as evidence that self-annotation can be used to create high quality gold standards.

3 Task definition

In this section, we describe the annotation task; give a definition of the relation types covered in this paper; and introduce the extraction model.

We focus on binary relations between two relation arguments occurring in the same sentence. To simplify the self-annotation process we restrict the first argument of the relation to the main entity of the Wikipedia article. As we are building text sensors for a context aware system, relations between geospatial entities are of interest. Thus we consider only relations that use a geospatial named entity as second argument.

We create the training set by automatically identifying all correct binary relations in the text. To this end, we extract the relations from the structured part of the Wikipedia, the infoboxes. Then we automatically find the corresponding

[1] http://opennlp.sourceforge.net/

sentences in the text and annotate the relations (see section 4). All other not yet marked binary relations between the main entity and geospatial entities are annotated as negative samples. The result of this step is a self-annotated training set.

In the second step of our task, the self-annotated training set is used to train the extraction model. The model only takes textual features as input and can be applied to any free text.

3.1 Classification task and relations used

Our relation extraction task is modeled as a classification task which considers a pair of named entities and decides whether they occur in the requested relation or not. The classifier uses extracted features for this decision. Features belong to three different classes. The first class contains *token-based features* and their linguistic labels like part-of-speech, lemma, stem. In the second class, we have *chunks* that aggregate one or more tokens into complex units. *Dependency relations* between the tokens are represented in the third class.

Our classifier is applicable to a wide spectrum of geospatial relation types. For the purposes of a focused evaluation, we selected two relations. The first type contains rivers and the bodies of water into which they flow. We call it *riverbodyOfWater* relation. Our second type is composed of relations between towns and the corresponding suburb. We call this *town-suburb* relation.

3.2 Wikipedia as resource

Wikipedia satisfies all corpus requirements for our task. It contains a lot of knowledge about geospatial data with unstructured (textual) and structured information. We consider only German Wikipedia articles because our target application is a German context aware system. In relation extraction for German, we arguably face more challenges – e.g., more complex morphology and freer word order – than we would in English.

For this work we consider only a subset of the German Wikipedia. We use all articles that belong to the following categories: Rivers by country, Mountains by country, Valleys by country, Islands by country, Mountain passes by country, Forests by country and Settlements by country.

For the annotation task we use the structural content of Wikipedia articles. Most articles belonging to the same categories use similar templates to represent structured information. One type of template is the infobox, which contains pairs of attributes and their values. These attribute-value pairs specify a wide range of geospatial relation types including fine-grained relations. In this work we consider only the infobox data and the article names from the structured data.

For context-aware systems fine-grained relation types are particularly relevant. Such relations are not represented in resources like DBPedia (Auer et al., 2007) or Yago (Suchanek et al., 2007) although they also consist of infobox data. Hence, we have to build our own extraction component (see section 5.2) when using infoboxes.

4 Self-Annotation

Self-annotation is a two-fold task. First, the structured data, in our case the infoboxes of Wikipedia articles, must be analyzed to get all relevant attribute-value pairs. Then all relevant geospatial entities are marked and extracted. In a second step these entities must be matched with the unstructured data.

In most cases, the extraction of the named entities that correspond to the required relations is trivial because the values in the infoboxes consist only of one single entity or one single link. But in some cases the values contain mixed content which can include links, entities and even free text. In order to find an accurate extraction method for those values we have developed several heuristics. See section 5.2 for discussion.

The second task links the extracted structured data to tokens in the textual data. Pattern based string matching methods are not sufficient to identify all relations in the text. In many cases, morphological rules need to be applied to identify the entities in the text. In other cases, the preprocessed text must be retokenized because the borders of multi-word expressions are not consistent with the extracted names in step one. One other issue is that some named entities are a subset of other named entities (*Lonau* vs. *kleine Lonau*;

Gollach

Die Gollach bei Aub-Baldersheim

Daten	
Lage	Bayern (Mittelfranken, Unterfranken), Deutschland
Gewässerkennzahl	DE: 2462
Länge	29,06 km
Quelle	zwischen den *Markt Nordheimer* Ortsteilen *Herbolzheim* und *Ulsenheim* am südwestlichen Rand des Steigerwaldes 49° 33′ 34″ N, 10° 18′ 43″ O
Quellhöhe	337,6 m
Mündung	bei Bieberehren (am Ende des Gollachtals) in die Tauber 49° 31′ 14″ N, 10° 0′ 2″ O

Figure 1: Infobox of the German Wikipedia article about *Gollach*.

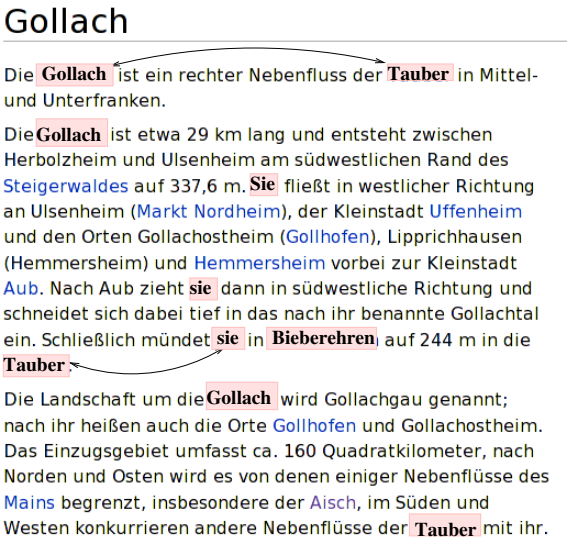
similar to *York* vs. *New York*). We have to use a longest match strategy to avoid such overlapping annotations.

The main goal of the self-annotation task is to reach the highest possible annotation quality. Thus, only complete extracted relations are used for the annotation process while incomplete data are excluded from the training set. This procedure reduces the noise in the labeled data.

4.1 Example

We use the river-bodyOfWater relation between the two rivers *Gollach* and *Tauber* to describe the self-annotation steps.

Figure 1 depicts a part of the infobox for the German Wikipedia article about the river *Gollach*. For this relation the attribute Mündung 'mouth' is relevant. The value contains unstructured information (i.e., text, e.g. *bei 'at' Bieberehren*) and structured information (the link from *Bieberehren* to its Wikipedia page). The relation we want to extract is that the river *Gollach* flows into the river *Tauber*.



Gollach

Die **Gollach** ist ein rechter Nebenfluss der **Tauber** in Mittel- und Unterfranken.

Die **Gollach** ist etwa 29 km lang und entsteht zwischen Herbolzheim und Ulsenheim am südwestlichen Rand des Steigerwaldes auf 337,6 m. **Sie** fließt in westlicher Richtung an Ulsenheim (Markt Nordheim), der Kleinstadt Uffenheim und den Orten Gollachostheim (Gollhofen), Lipprichhausen (Hemmersheim) und Hemmersheim vorbei zur Kleinstadt Aub. Nach Aub zieht **sie** dann in südwestliche Richtung und schneidet sich dabei tief in das nach ihr benannte Gollachtal ein. Schließlich mündet **sie** in **Bieberehren** auf 244 m in die **Tauber**.

Die Landschaft um die **Gollach** wird Gollachgau genannt; nach ihr heißen auch die Orte Gollhofen und Gollachostheim. Das Einzugsgebiet umfasst ca. 160 Quadratkilometer, nach Norden und Osten wird es von denen einiger Nebenflüsse des Mains begrenzt, insbesondere der Aisch, im Süden und Westen konkurrieren andere Nebenflüsse der **Tauber** mit ihr.

Figure 2: Textual content of the German Wikipedia article about *Gollach*. All named entities which are relevant for the river-bodyOfWater relation are highlighted. This article contains two instances for the relation between *Gollach* and *Tauber*.

Figure 2 shows the textual content of the *Gollach* article. We have highlighted all relevant named entities for the self-annotation process. This includes the name of the article and instances of the pronoun *sie* referring to *Gollach*. Our matching algorithm identifies two sentences as positive samples for the relation between *Gollach* and *Tauber*:

- (i) Die *Gollach* ist ein rechter Nebenfluss der *Tauber* in Mittel- und Unterfranken. (The *Gollach* is a right tributary of the *Tauber* in Middle and Lower Franconia.)
- (ii) Schließlich mündet *sie* in Bieberehren auf 244 m in die *Tauber*. (Finally, *it* discharges in Bieberehren at 244 m above MSL into the *Tauber*.)

5 Processing

In this section we describe how the self-annotation method and relation extraction is implemented. First we introduce the interaction with the Wikipedia resource to acquire the structured and unstructured information for the processing

pipeline. Second we present the components of the UIMA pipeline which are used for the relation extraction task.

5.1 Wikipedia interaction

We use the JWPL API (Zesch et al., 2008) to pre-process the Wikipedia data. This interface provides functions to extract structured and unstructured information from Wikipedia. However, many Wikipedia articles do not adhere to valid Wikipedia syntax (missing closing brackets etc.). The API also does not correctly handle all Wikipedia syntax constructions. We therefore have enhanced the API for our extraction task to get high quality data for German Wikipedia articles.

5.2 Infobox extraction

As discussed in section 4 infoboxes are the key resource for the self-annotation step. However the processing of infoboxes that include attribute-value pairs with mixed content is not trivial.

For each new relation type an initial manual effort is required. However, in comparison to the complete annotation of a training corpus, this effort is small. First the attributes used in the infoboxes of the Wikipedia articles relevant for a specific relation have to be analyzed. The results of this analysis simplify the choice of the correct attributes. Next, the used values of these attributes must be investigated. If they contain only single entries (links or named entities) the extraction is trivial. However, if they consist of mixed content (see section 4.1) then specific extraction methods have to be applied. We investigated different heuristics for the self-annotation process to get a method that can easily be adapted to new relation types.

Our first heuristic includes a set of rules specifying the extraction of the values from the infoboxes. This heuristic gives an insufficient basis for the self-annotation task because the rich morphology and free word order in German can not be modeled with simple rules. Moreover, handcrafted rules are arguably not as robust and maintainable as a statistical classifier trained on self-annotated training material.

Our second heuristic is a three step process. In step one we collect all links in the mixed content and replace them by a placeholder. In the second step we tag the remaining content with the OpenNLP tokenizer to get all named entities. Both collected lists are then looked up in a lexicon that contains named entities and the corresponding geospatial classes. This process requires a normalization procedure that includes the application of morphological methods. The second method can be easily adapted to new relation types.

5.3 UIMA

The self-annotated corpora are processed by several components of the UIMA (Müller et al., 2008) pipeline. The advantage of exchangeable collection readers is that they seamlessly handle structured and unstructured data. Another advantage of using UIMA is the possibility to share components with other research groups. We can easily exchange different components, like the usage of the commonly known OpenNLP processing tools or the FSPar NLP engine (Schiehlen, 2003) (which includes the TreeTagger (Schmid, 1995)). This allows us to experiment with different approaches, e.g., shallow vs. deep analysis. The components we use provide linguistic analysis on different levels: tokens, morphology, part of speech (POS), chunking and partial dependency analysis. Figure 4 shows the results after the linguistic processing of our sample sentence. For this work only a few annotations are wrapped as UIMA types: token (incl. lemma, POS), multiword, sentence, NP, PP and dependency relations (labeled edges between tokens). We will introduce our machine learning component in section 5.5. Finally, the CAS consumers allow us to store extracted facts in a context model.

Figure 3 shows the article about *Gollach* after linguistic processing. In the legend all annotated categories are listed. We highlighted all marked relations, all references to the article name (referred to as subject in the figure) and links. After selection of the *Tauber* relation, all annotations for this token are listed in the right panel.

5.4 Coreference resolution

Using anaphora to refer to the main entity is a common practice of the authors of Wikipedia ar-

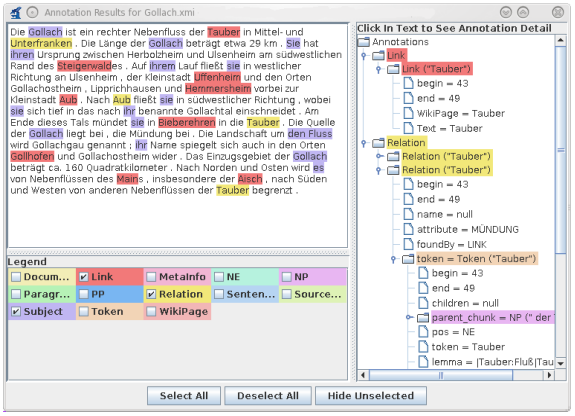


Figure 3: Screenshot of the UIMA Annotation-Viewer.

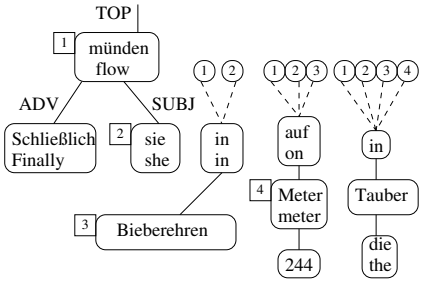


Figure 4: Dependency parser output of the FSPar framework.

	linguistic effort	description
F1	pos-tagging	window size 3, LEMMA
F2	chunk-parse	parent chunks
F3	dependency-parse	dependency paths betw. NEs

Table 1: List of feature types

ticles. Coreference resolution is therefore necessary for our annotation task. A shallow linguistic analysis showed that the writing style is similar throughout Wikipedia articles. Based on this observation, we empirically investigated some geospatial articles and came to the conclusion that a simple heuristic is sufficient for our coreference resolution problem. In almost all articles, pronouns refer to the main entity of the article. In addition we include some additional rules to be able to establish coreference of markables such as *der Fluss* 'the river' or *der Bach* 'the creek' with the main entity.

5.5 Supervised relation extraction

We use the ClearTK (Ogren et al., 2008) toolkit, which is also an UIMA component, for the relation extraction task. It contains wrappers for different machine learning suites. Our initial experiments showed that the MaximumEntropy classifier achieved the best results for our classification task. The toolkit provides additional extensible feature methods. Because we view self-annotation and fine-grained named entity recognition as our main contributions, not feature selection, we only give a brief overview of the features we use.

F1 is a window based bag-of-words feature (window size = 3). It considers lemma and part-of-speech tag of the tokens. F2 is a phrase based extractor that uses the parent phrase of both entities (max 2 levels). F3 is a representation of all possible dependency paths between the article's main entity and a target entity, where each path is represented as a feature vector. In most cases, more than one path is returned by the partial dependency parser (which makes no disambiguation decisions) and included in the feature representation. Figure 4 depicts the dependency parser output of our sample sentence. Each pair of square and circle with the same number corresponds to one dependency. These different possible dependency combinations give rise to 8 possible paths between the relation entities *Tauber* and *sie 'she'* although our example sentence is a very simple sentence.

6 Evaluation

We evaluate the system in two experiments. The first considers the relation between suburbs and their parent towns. In the second experiment the river-bodyOfWater relation is extracted. The experiments are based on the previously described extracted Wikipedia corpus. For each experiment a new self-annotated corpus is created that is split into three parts. The first part (60%) is used as training corpus. The second part (20%) is used as development corpus. The remaining 20% is used for the final evaluation and was not inspected while we were developing the extraction algorithms.

6.1 Metric used

Our gold standard includes all relations of each article. Our metric works on the level of type and is independent of how often the same relation occurs in the article. The metric counts a relation as true positive (TP) if the system extracted it at least once. If the relation was not found by the system a false negative (FN) is counted. A false positive (FP) is given if the system extracts a relation between two entities that is not part of the (infobox-derived) gold standard for the article. All three measures are used to calculate precision ($P = \frac{TP}{TP+FP}$), recall ($R = \frac{TP}{TP+FN}$), and F_1-score ($F_1 = 2\frac{P*R}{P+R}$).

6.2 Town-suburb extraction

The town-suburb extractor uses one attribute of the infobox to identify the town-suburb relation. There is no schema drift in the infobox data and the values contain only links. Therefore the self-annotation works almost perfectly. The only exceptions are articles without an infobox which cannot be used for training. However, this is not a real issue because the amount of remaining data is sufficient: 9000 articles can be used for this task. The results in table 2 show that the classifier that uses F1, F2 and F3 (that is, including the dependency features) performs best.

engine	features	F_1	recall	precision
FSPar	F1	64.9	79.0%	55.7%
FSPar	F1, F2	89.6	90.2%	89.5%
FSPar	F1, F2, F3	98.3	98.8%	97.8%

Table 2: Results of different feature combinations on the test set for town-suburb relation

6.3 River-bodyOfWater extraction

For the extraction of the river-bodyOfWater relation the infobox processing is more difficult. We have to handle more attributes because there is schema drift between the different users. It is hence necessary to merge information coming from different attribute values. The other difficulty is the usage of mixed contents in the values. Another main difference to the town-suburb relation is that the river-bodyOfWater relation is often not mentioned in the first sentence (which usually gives a short definition about the the main entity). Thus, the self-annotation method has to deal with the more complex sentences that are common later in the article. This also contributes to a more challenging extraction task.

Our river-bodyOfWater relation corpus consists of 3000 self-annotated articles.

Table 3 shows the performance of the extractor using two different linguistic components as described in section 5.3. As in the case of town-suburb extraction the classifier that uses all features, including dependency features, performs best.

engine	features	F_1	recall	precision
FSPar	F1	51.8%	56.6%	47.8%
FSPar	F1,F2	72.1%	68.9%	75.7%
FSPar	F1,F2,F3	78.3%	74.1%	83.0%
OpenNLP	F1	48.0%	62.8%	38.8%
OpenNLP	F1,F2	73.3%	71.7%	74.7%

Table 3: Results of different feature combinations on the test set for river-bodyOfWater extraction

6.4 Evaluation of self-annotation

To evaluate the quality of self-annotation, we randomly selected one set of 100 self-annotated articles from each data set and labeled these sets manually. These annotations are used to calculate the inter-annotator agreement between the human annotated and machine annotated instances. We use Cohen's κ as measure and get a result of 1.00 for the town-suburb relation. For the river-bodyOfWater relation we got a κ-value of 0.79, which also indicates good agreement.

We also use a gazetteer to evaluate the quality of all town-suburb relations that were extracted for our self-annotated training set. The accuracy is nearly perfect (only one single error), which is good evidence for the high quality of Wikipedia.

Required size of self-annotated training set. The performance of a supervised system depends on the size of the training data. In the self-annotation step a minimum of instances has to be annotated, but it is not necessary to self-annotate all available articles.

We reduced the number of articles used in the training size to test this hypothesis. Reducing the entire training set of 9000 (respectively, 3000) self-annotated articles to 1000 reduces F1

by 2.0% for town-suburb and by 2.4% for river-bodyOfWater; a reduction to 100 reduces F1 by 8.5% for town-suburb and by 9.3% for river-bodyOfWater (compared to the 9000/3000 baseline).

7 Discussion

Wu and Weld (2007) observed schema drift in their work: Wikipedia authors do not not use infobox attributes in a consistent manner. However, we did not find schema drift to be a large problem in our experiments. The variation we found can easily be handled with a small number of rules. This can be due to the fact that the quality of Wikipedia articles improved a lot in the last years through the introduction of automatic maintenance tools like bots[2]. Nevertheless, the development of self-annotation for a new relation type requires some manual work. The developer has to check the quality of the extraction relations in the infoboxes. This can lead to some additional adaptation work for the used attributes such as merging or creating rules. However, a perfect coverage is not required because the extraction system is only used for training purposes; we only need to find a sufficiently large number of positive training instances and do not require exhaustive labeling of all articles.

It is important to note that considering partially found relations as negative samples has to be avoided. Wrong negative samples have a generally unwanted impact on the performance of the learned extraction model. A developer has to be aware of this fact. In one experiment, the learned classifiers were applied to the training data and returned a number of false positive results – 40 in case of the river-bodyOfWater relation. 31 of these errors were not actual errors because the self-annotation missed some true instances. Nevertheless, the trained model recognizes these samples as correct; this could perhaps be used to further improve the quality of self-annotation.

Manually labeled data also includes noise and the benefit of self-annotation is substantial when the aim is to build a fine-grained relation extraction system in a fast and cheap way.

The difference of the results between OpenNLP and FSPar engines are smaller than expected. Although sentence splitting is poorly done by OpenNLP the effect on the extraction result is rather low. Another crucial point is that the lexicon-based named entity recognizer of the FSPar engine that was optimized for named entities used in Wikipedia has no significant impact on the overall performance. Thus, a basic set of NLP components with moderate error rates may be sufficient for effective self-annotation.

8 Conclusion

This paper described a new approach to developing and implementing a complete system to extract fine-grained geospatial relations by using a supervised machine learning approach without expensive manual labeling. Using self-annotation, systems can be rapidly developed and adapted for new relations without expensive manual annotation. Only some manual work has to be done to find the right attributes in the infoboxes. The matching process between infoboxes and text is not in all cases trivial and for some attributes additional rules have to be modeled.

9 Acknowledgment

This project was funded by DFG as part of Nexus (Collaborative Research Centre, SFB 627).

References

Auer, Sören, Christian Bizer, Georgi Kobilarov, Jens Lehmann, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *In 6th Intl Semantic Web Conference, Busan, Korea*, pages 11–15. Springer.

Blessing, Andre, Stefan Klatt, Daniela Nicklas, Steffen Volz, and Hinrich Schütze. 2006. Language-derived information and context models. In *Proceedings of 3rd IEEE PerCom Workshop on Context Modeling and Reasoning (CoMoRea) (at 4th IEEE International Conference on Pervasive Computing and Communication (PerCom'06))*.

Dürr, Frank, Nicola Hönle, Daniela Nicklas, Christian Becker, and Kurt Rothermel. 2004. Nexus–a platform for context-aware applications. In Roth, Jörg,

[2] See en.wikipedia.org/wiki/Wikipedia:Bots. The edit history of many articles shows that there is a lot of automatic maintenance by bots to avoid schema drift.

editor, *1. Fachgespräch Ortsbezogene Anwendungen und Dienste der GI-Fachgruppe KuVS*, pages 15–18, Hagen, Juni. Informatik-Bericht der FernUniversität Hagen.

Hahn, Udo, Ekaterina Buyko, Rico Landefeld, Matthias Mühlhausen, Michael Poprat, Katrin Tomanek, and Joachim Wermter. 2008. An overview of JCoRe, the JULIE lab UIMA component repository. In *Proceedings of the LREC'08 Workshop 'Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP'*, Marrakech, Morocco, May.

Jiang, Jing. 2009. Multi-task transfer learning for weakly-supervised relation extraction. In *ACL-IJCNLP '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*, pages 1012–1020, Morristown, NJ, USA. Association for Computational Linguistics.

Müller, Christof, Torsten Zesch, Mark-Christoph Müller, Delphine Bernhard, Kateryna Ignatova, Iryna Gurevych, and Max Mühlhäuser. 2008. Flexible uima components for information retrieval research. In *Proceedings of the LREC 2008 Workshop 'Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP',* Marrakech, Morocco, May 31, 2008. 24–27.

Nothman, Joel, Tara Murphy, and James R. Curran. 2009. Analysing wikipedia and gold-standard corpora for ner training. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 612–620, Morristown, NJ, USA. Association for Computational Linguistics.

Ogren, Philip V., Philipp G. Wetzler, and Steven Bethard. 2008. Cleartk: A uima toolkit for statistical natural language processing. In *UIMA for NLP workshop at Language Resources and Evaluation Conference (LREC)*.

Schiehlen, Michael. 2003. Combining deep and shallow approaches in parsing german. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 112–119, Morristown, NJ, USA. Association for Computational Linguistics.

Schmid, Helmut. 1995. Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

Suchanek, Fabian M., Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *16th international World Wide Web conference (WWW 2007)*, New York, NY, USA. ACM Press.

Wu, Fei and Daniel S. Weld. 2007. Autonomously semantifying wikipedia. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007*, pages 41–50.

Zesch, Torsten, Christof Müller, and Iryna Gurevych. 2008. Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*.

Zhang, Ziqi and José Iria. 2009. A novel approach to automatic gazetteer generation using wikipedia. In *People's Web '09: Proceedings of the 2009 Workshop on The People's Web Meets NLP*, pages 1–9, Morristown, NJ, USA. Association for Computational Linguistics.