Fine-Grained Geographical Relation Extraction from Wikipedia

Andre Blessing, Hinrich Schütze

Institute for Natural Language Processing
Universität Stuttgart
Germany

Abstract

In this paper, we present work on enhancing the basic data resource of a context-aware system. First, we introduce a supervised approach to extracting geographical relations on a fine-grained level. Second, we present a novel way of using Wikipedia as a corpus based on self-annotation. A self-annotation is an automatically created high-quality annotation that can be used for training and evaluation. The fine-grained relations are used to complete gazetteer data. The precision and recall scores of more than 97% confirm that a statistical IE pipeline can be used to improve the data quality of community-based resources.

1. Introduction

In the last years, linguistic resources have become more important in new domains like context-aware systems. For a system like NEXUS ( Dürr et al., 2004), which is based on a context model, geospatial resources can be viewed as the backbone. These resources must be of high quality to achieve broad adoption by users of a system like NEXUS. To create such high-quality resources, new NLP methods are needed. (Blessing et al., 2006) introduced the idea of a text-sensor to acquire new information for a context model by analyzing textual data.

Electronic text offers a wealth of information about geospatial data and can be used to improve the completeness and accuracy of geospatial resources (e.g., gazetteers). The community-based GeoNames project1 is such a resource. Our first contribution in this paper is to assist the GeoNames project by providing relations between urban entities that will be extracted from electronic text.2 The currently used heuristics in GeoNames retrieve many incorrect part-of relations between suburbs, municipalities and counties. The user community has asked repeatedly for more accurate part-of relations between the administrative levels, demonstrating the importance of the problem. Such data resources are an important source for other tasks like Geo-Tagging (Blessing et al., 2007).

Most work on geospatial information extraction (IE) targets English text. We are building a system for German, a much more challenging language for IE (freer word order, varied compounds and harder named entity recognition because all nouns are uppercase). As a consequence, pattern based approaches have very limited success for German.

Wikipedia can be an important source for developing language resources by means of self-annotation – using structured data to create high-quality annotations automatically. (Nothman et al., 2009) showed that such an annotated Wikipedia corpus can be used as gold standard for NER training.

1http://www.geonames.org

2GeoNames models Germany by 4 administrative levels: state (3, Bundesland) – county (2, Kreis) – municipality (1, Gemeinde) – suburb (0, part of municipality)

2. Task Definition

We address the task of extracting the two geographic relations $R_{0-1}$ and $R_{1-2}$ from Wikipedia. Two examples from sentences (iii) and (iv) below are:

- (i) $R_{1-2}$(Gebroth, Bad Kreuznach)
- (ii) $R_{0-1}$(Sohlbach, Netphen)

$R_{0-1}$ links each suburb or district (‘Orts-/Stadtteile’, level 0 of our hierarchy) to the municipality or city (‘Gemeinde’, level 1) it is part of. $R_{1-2}$ links each municipality (‘Gemeinde’, level 1) to the county (‘Landkreis’, level 2) it is part of. We use municipality as a technical term in this paper. In particular, a suburb/district is not a municipality. Sentence (iii) states that (i) is true and sentence (iv) states that (ii) is true. Named entities (which are potential candidates for relations) are italicized.

- (iii) $R_{1-2}$: Gebroth ist eine Ortsgemeinde im Landkreis Bad Kreuznach in Rheinland-Pfalz (Deutschland).

(Gebroth is a municipality in the county Bad Kreuznach in Rheinland-Pfalz (Germany).)

- (iv) $R_{0-1}$: Sohlbach ist ein Stadtteil von Netphen im Kreis Siegen-Wittgenstein in Nordrhein-Westfalen mit 143 Einwohnern.

(Sohlbach is a suburb of Netphen in the county Siegen-Wittgenstein in Nordrhein-Westfalen with 143 inhabitants.)

We formalize the relation retrieving task as a multiclass classification problem that discriminates between three classes: $R_{0-1}$, $R_{1-2}$ and a third class that includes all other possible binary relations between entities. Examples for the third class in (iv) are $R$(Sohlbach,Siegen-Wittgenstein) (a suburb/district-county relationship that could easily be misrecognized as a municipality-county relationship) and $R$(Sohlbach,Nordrhein-Westfalen) (suburb/district-state relationship).

IE for part-of relations is not new (Culotta et. al., 2006). However, our task (defined by GeoNames and the needs of

3This sentence will be used as example in the remaining paper.
context-aware systems like NEXUS) is more complex since we need to distinguish between two part-of relations that differ only in level of hierarchy, a very subtle difference.

3. Wikipedia self-annotation

Wikipedia is a large collaborative encyclopedia. It is a useful resource for our work because it contains two types of different context: (i) unstructured text and (ii) structured data: templates (e.g., infoboxes about cities), lists and tables. One advantage of using Wikipedia as data source is that our requested relations are not only stored in unstructured text, but are also included in structured data templates. For our purposes, the templates Infobox Gemeinde in Deutschland (German municipality) and Infobox Ortsteil einer Gemeinde (German suburb) provide information on \( R_0 - 1 \) and \( R_1 - 2 \) in a well-defined format. An important contribution of this paper is that we show how structured information like infoboxes in Wikipedia can be used to generate self-annotations – which are then available for training and evaluating statistical classifiers. Figure 1 shows how the \( R_1 - 2 \) (Gebroth, Bad Kreuznach) relation is annotated by using structured information of the infobox and the article name. We used JWPL (Java Wikipedia Library (Zesch et al., 2008)) to extract all articles of the German Wikipedia about municipalities and suburbs/districts. 9037 articles met our criteria about completeness of the infoboxes and the integrity of the first sentence (main entities of the infobox must be mentioned in first sentences of the article). The 9037 first sentences of the articles are concatenated as a corpus. In the next step the structured information is used to annotate the unstructured textual corpus with the two relations defined above. We call this step self-annotation because no manual work is needed. To support the supervised approach we split this annotated corpus into three parts to enable a clean evaluation. The first 60% (5357 sentences) are used as the training set during development. The next 20% (1840 sentences) are used for the evaluation in the development phase. The remaining 20% (1840 sentences) are the test set and used for evaluation.

4. UIMA Pipeline

The above defined corpus is processed by several components of a UIMA (Hahn et al., 2008) pipeline. Our main analysis engine is a wrapper around the FSPar
NLP engine (Schiehlen, 2003) (which includes the TreeTagger). This engine provides linguistic analysis on different levels (tokenizer, morphology, part of speech (POS), chunking and partial dependency analysis). For this work only a few annotations are wrapped as UIMA types: token (incl. lemma, POS), multi-word, sentence, NP, PP and dependency relations (labeled edges between tokens).

Figure 2: Output of the FSPar dependency parser.

A lexicon is used to mark all named entities in the first sentence. We use heuristics to address spelling errors and variants (which occur frequently). Table 1 shows our sample sentence including POS and lemma tags. 4 named entities are found (NE₁...NE₄). All possible binary relations are build (R(NE₁,NE₂),R(NE₁,NE₃),...) for the classification step. Figure 2 depicts the output of the FSPar dependency parser. One disadvantage of the parser is that no disambiguation step is included. In our example the “in” token has no unambiguous parent node and the parser returns all possible dependency relations.

Another component of our pipeline is the ClearTk (Ogren et al., 2008) framework. We extended its feature extraction methods and use its classification framework. We use only the OpenNLP-MaxEnt algorithm because it performs best on the development set.

5. Feature Design

Table 2 introduces our features. The second column shows which linguistic processing is necessary to calculate the feature.

Table 2: List of feature types

| linguistic effort | description |
|-------------------|-------------|
| F0                | none        |
| F1                | pos-tagging |
| F2                | chunk-parse |
| F3                | dependency-parse |

F0 is our base feature that needs no linguistic analysis. It stores information about the distance between the two entities and the position of the target entity. F1 is a window based feature (window size = 2) that considers lemma and POS information. F2 is calculated on the basis of parent chunks (max 2 levels). F3 stores all possible dependency paths (each path is represented as a feature vector) between the subject entity and target entity. In most cases more than one path is stored because the partial dependency parser makes no disambiguation decisions. The parser also recognizes the fields of the German sentence (Vorfeld, Mittelfeld, Nachfeld), its main structural elements. We exploit this and store all words inside the right sentence bracket of the field model in F3 to get more information about the main verb.

6. Evaluation

For the evaluation of the self-annotation we used the annotated \( R_{1 \rightarrow 2} \) relations in the corpus and compared them with data of the Federal Statistical Office of Germany. The advantage of this method is that we can prove the quality of the knowledge base (infoboxes) and the quality of annotation in one step. We got a successful result with an accuracy of 99.9% (1 error in 1304 sentences).

We use precision\(^4\) and recall to evaluate the classifier on the test set.

Table 3 shows the application of different feature combinations on the test set

Table 3: Results of different feature combinations on the test set

| Classifier | features | precision | recall | FF | FN |
|-----------|----------|-----------|--------|----|----|
| 1         | F0       | 79.0%     | 55.7%  | 279| 833|
| 2         | F0+F1    | 92.4%     | 89.3%  | 138| 202|
| 3         | F0+F2    | 90.2%     | 89.5%  | 182| 198|
| 4         | F0+F3    | 97.7%     | 97.4%  | 43 | 48 |
| 5         | F0+F3    | 98.8%     | 97.8%  | 23 | 41 |

\(^4\) Correctly classified instances of \( R_{0 \rightarrow 1} \) and \( R_{1 \rightarrow 2} \) are true positive (TP), unclassified instances are false negative (FN) and misclassified instances are false positive (FP).
4. (TP) Hürthgenwald ist eine Gemeinde in Nordrhein-Westfalen, Deutschland und gehört zum Kreis Düren. (Hürthgenwald ist ein Stadtteil) (Hürthgenwald ist eine Stadt) (Hürthgenwald ist ein Teil der Stadt Düren).

Sentences 1 and 2 state that a suburb was a municipality, but no longer is. In the first case only the word ehemals ‘formerly’ indicates that fact and is not classified correctly. In the second case the past tense of the main verb indicates the “past” meaning and is correctly classified. Sentence 3 shows that coordinations are sometimes not handled correctly by the classifier. Sentence 4 is an example of a difficult coordination (large distance between elements of the relation) being handled correctly.

7. Related Work

(Wu and Weld, 2007) used the term “autonomously Semantifying Wikipedia” to describe their approach. They augmented infoboxes by a bootstrapping method. For this text and other structured information is used to complete missing data. We differ by the used target language (German) which raises new challenges and by using the infobox data to annotate textual content for further research. (Zhang and Iria, 2009) introduced a method to automatically generate gazetteers from seed lists using Wikipedia. In difference to our work their method uses textual and structural content for the extraction. They also do not distinguish between fine-grained named entity classes. (Mika et al., 2008) considered the problem of semantic annotation of Wikipedia in other way. As knowledge base for the annotation process they used DBPedia that is derived from structured Wikipedia data.

8. Outlook

Wikipedia provides more information than we have used so far. In the future we will consider additional structured data such as links and categories to model more relations. We believe that this type of self-annotated corpus will be very significant for future IE resource development.

9. Conclusion

In this paper, we presented work on enhancing the basic data resource of a context-aware system. First, we introduced a supervised approach to extracting geographical relations on a fine-grained level. Second, we presented a novel way of using Wikipedia as a corpus based on self-annotation. A self-annotation is an automatically created high-quality annotation that can be used for training and evaluation. The fine-grained relations are used to complete gazetteer data. The precision and recall scores of more than 97% confirmed that a statistical IE pipeline can be used to improve the data quality of community-based resources.

10. References

Andre Blessing, Reinhard Kunz, and Hinrich Schütze. 2007. Towards a context model driven German geotagging system. In GIR ’07: Proceedings of the 4th ACM workshop on geographical information retrieval, pages 25–30, New York, NY, USA. ACM.

Frank Dürr, Nicola Höhle, Daniela Nicklas, Christian Becker, and Kurt Rothermel. 2004. Nexus—a platform for context-aware applications. In Jörg Roth, editor, 1. Fachgespräch Ortsbezogene Anwendungen und Dienste der GI-Fachgruppe KuVS, pages 15–18, Hagen, Juni. Informatik-Bericht der FernUniversität Hagen.

Peter Mika, Massimiliano Ciaramita, Hugo Zaragoza, and Jordi Atserias. 2008. Learning to tag and tagging to learn: A case study on wikipedia. IEEE Intelligent Systems, 23(5):26–33.

Jordi Atserias. 2008. Learning to tag and tagging to learn: A case study on wikipedia. IEEE Intelligent Systems, 23(5):26–33.

Michael Schiehlen. 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.

Fei Wu and Daniel S. Weld. 2007. Autonomously semantifying wikipedia. In CIKM ’07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, pages 41–50, New York, NY, USA. ACM.

Torsten Zesch, 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).

Ziqi Zhang 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.