The TIGER Corpus Navigator

Sebastian Hellmann[1] Jörg Unbehauen[1]
Christian Chiarcos[2] Axel-Cyrille Ngonga Ngomo[1]

[1]University of Leipzig [2]University of Potsdam
E-mail: chiarcos@uni-potsdam.de
{hellmann|unbehauen|ngonga}@informatik.uni-leipzig.de

Abstract

Linguistically annotated corpora are a central resource in NLP. The extraction of formal knowledge from these corpora, however, is a tedious process. We introduce the Tiger Corpus Navigator, a Semantic Web system which aids users to classify and retrieve sentences from linguistic corpora – here, the Tiger corpus – on the basis of abstract linguistic concepts.

These linguistic concepts are specified extensionally, thus, independent from the underlying annotation: The user provides a small set of pre-classified sentences that represent instances (positive examples) or counterinstances (negative examples) of the corresponding concept, and the system automatically acquires a formal OWL/DL specification of the underlying concept using an Active Machine Learning approach.

1 Introduction

A large number of annotated corpora have become available over the past years. Still, the retrieval of dedicated linguistic knowledge for given applications or research questions out of these corpora remains a tedious process. An expert in linguistics might have a very precise idea of the concepts she would like to retrieve from a corpus. Yet, she faces a number of challenges when trying to retrieve corresponding examples out of a particular corpus:

access she needs a tool that is able to process the format of the corpus, that is easy to deploy, and that provides an intuitive user interface

documentation she needs to be familiar with the annotations and the query language

representation she needs a representation of the results so that these can be studied more closely or that they can be processed further with other NLP tools.

In this paper, we describe a novel approach to this problem that starts from the premise that linguistic annotations can be represented by means of existing standards developed in the Semantic Web community: RDF and OWL[1] are well-suited for data integration, and they allow to represent different corpora and tagsets in a uniform way.

We present the Tiger Corpus Navigator, an Active Machine Learning tool that allows a user to extract formal definitions of extensionally defined concepts and the corresponding examples out of annotated corpora. Based on an initial seed of examples provided by the user, the Navigator learns a formal OWL Class Definition of the concept that the user is interested in. This definition is converted into a SPARQL query[2] and passed to Virtuoso,[3] a triple store database with reasoning capabilities. The results are gathered and presented to the user to choose more examples, to refine the query, and to improve the formal definition. The data basis for the Navigator is an OWL/RDF representation of the Tiger corpus[4] and a set of ontologies that represent its linguistic annotations.

Our tool, available under `http://tigernavigator.nlp2rdf.org`, addresses and circumvents the barriers to the acquisition of knowledge out of corpora presented above:

(i) it does not need any deployment and provides a user interface in a familiar surrounding, the browser,

(ii) the concept descriptions acquired during the classifier refinement represent the (conceptual representation of the) annotations in the corpus in an explicit and readable way, and finally,

(iii) the Navigator uses OWL; the query results are thus represented in a readable, portable and sustainable way.

2 Tools and Resources

Several categories of tools and resources need to be integrated to enable the implementation of the goals presented above: We employ the **DL-Learner** [16] to learn class definitions for linguistic concepts; **NLP2RDF** [12] is applied for the conversion and ontological enrichment of corpus data; and the **OLiA ontologies** [5] provide linguistic knowledge about the annotations in the corpus.

2.1 DL-Learner

The DL-Learner extends Inductive Logic Programming to Descriptions Logics, OWL and the Semantic Web; it provides a OWL/DL-based machine learning tool

[1] `http://www.w3.org/TR/rdf-concepts`, `http://www.w3.org/TR/owl-ref`
[2] `http://www.w3.org/TR/rdf-sparql-query`
[3] `http://virtuoso.openlinksw.com`
[4] `http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERCorpus`

to solve supervised learning tasks and support knowledge engineers in constructing knowledge. The induced classes are short and readable and can be stored in OWL and reused for classification. OWL/DL is based on Description Logics that can essentially be understood as fragments of first-order predicate logic with less expressive power, but usually decidable inference problems and a user-friendly variable free syntax. OWL Class definitions form a subsumption hierarchy that is traversed by DL-Learner starting from the top element (*owl:Thing*) with the help of a refinement operator and an algorithm that searches in the space of generated classes. An example of such a refinement chain is (in Manchester OWL Syntax):

(Sentence) $\rightsquigarrow$
(Sentence and hasToken some Thing) $\rightsquigarrow$
(Sentence and hasToken some VVPP) $\rightsquigarrow$
(Sentence and hasToken some VVPP and hasToken some (stts:AuxiliaryVerb and hasLemma value "werden"))

The last class can easily be paraphrased into: A sentence that has (at least) one Token, which is a past participle (*VVPP*), and another Token, which is an AuxiliaryVerb with the lemma *werden* (passive auxiliary, lit. 'to become'). Detailed information can be found in [16] and under `http://dl-learner.org`.

2.2 NLP2RDF

NLP2RDF[5] is a framework that integrates multiple NLP tools in order to assess the meaning of the annotated text by means of RDF/OWL descriptions: Natural language (a character sequence) is converted into a more expressive formalism – in this case OWL/DL – that grasps the underlying meaning and serves as input for (high-level) algorithms and applications.

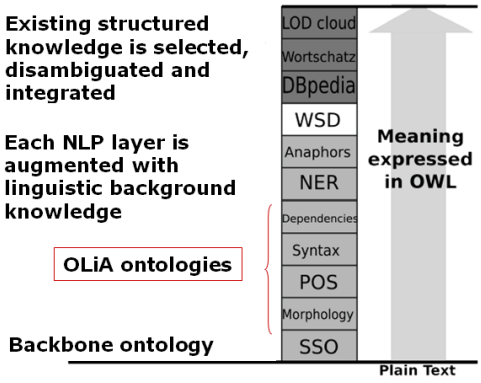

Figure 1: NLP2RDF stack

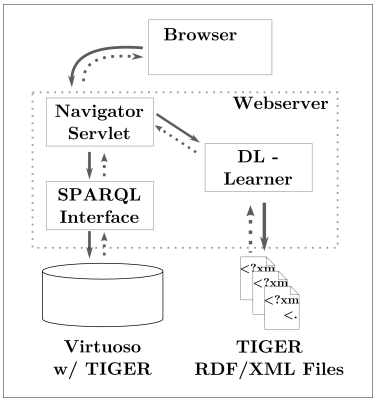

Figure 2: Architecture of the Tiger Corpus Navigator

[5] `http://nlp2rdf.org`, `http://code.google.com/p/nlp2rdf`

In a first step, sentences are tokenized and aggregated in a *Structured Sentence ontology (SSO)*. The SSO consists of a minimal vocabulary that denotes the basic structure of the sentence such as tokens and relative position of a token in a sentence.

As shown in Fig. 1, the SSO serves as the backbone model, which is then augmented additional layers of annotations:

(1) features from NLP tools
in light grey: morphology, parts of speech (POS), syntactic structures and edge labels (syntax, dependencies), named entity recognition (NER), coreference (anaphors)

(2) rich linguistic ontologies for these features (Sect. 2.3)
combined in a *tagset-ontology pair* for every level mentioned in (1)

(3) background knowledge from the Web of Data
examples in dark grey: Linking Open Data (LOD) Cloud,[6] DBPedia,[7] and Wortschatz[8]

(4) additional knowledge
knowledge created by the Navigator (Sect. 2.1) or derived from the steps described above (e.g., in white: word sense disambiguation, WSD)

2.3 Linguistic Ontologies

The Ontologies of Linguistic Annotations [5, OLiA] represent an architecture of modular OWL/DL ontologies that formalize several intermediate steps of the mapping between concrete annotations, a Reference Model and external terminology repositories, such as GOLD[9] or the ISO TC37/SC4 Data Category Registry:[10]

- Multiple Annotation Models formalize annotation schemes and tag sets, e.g., STTS for the part of speech tags of the Tiger corpus.
- The Reference Model provides the integrating terminology for different annotation schemes (OLiA Annotation Models).
- For every Annotation Model, conceptual subsumption relationships between Annotation Model concepts and Reference Model concepts are specified in a Linking Model. Other Linking Models specify relationships between Reference Model concepts and external terminology repositories [6].

[6] http://richard.cyganiak.de/2007/10/lod
[7] http://dbpedia.org
[8] http://wortschatz.uni-leipzig.de
[9] http://linguistics-ontology.org
[10] http://www.isocat.org

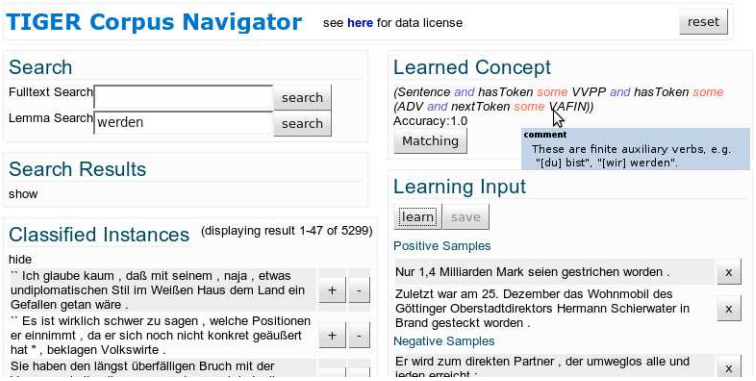

Figure 3: Screenshot of the Tiger Corpus Navigator

For the current paper, we focused on the STTS Annotation Model[11] that covers the morphosyntactic annotations in the Tiger corpus.

The usage of OLiA combined with NLP2RDF offers two major advantages: OLiA provides a growing collection of annotation models for more than 50 languages, that are interlinked with the OLiA Reference Model (and further to community-maintained repositories of linguistic terminology). The adaption of the Navigator to other corpora and other languages is thus easily possible. The interlinking further allows to reuse learned classes on other corpora and even to learn on a combination of different corpora.

3 The Tiger Corpus Navigator

Figure 2 shows the architecture of the Tiger Corpus Navigator: The Virtuoso triple store contains the whole corpus in RDF and allows queries over the complete data for retrieval, the data used by DL-Learner consists of one file for the OWL schema and 50,474 RDF/XML files (one per sentence), which it loads on demand according to the given examples.

With the Navigator user interface (Fig. 3), the user starts his research by searching for sentences with certain lemmas or words. The retrieved sentences are presented on the left side. They can be moved to the right panel and classified as positive or negative examples, i.e., as instances or counterinstances of the target concept. Upon pressing the *Learn* button, they are sent to the DL-Learner and the learned OWL Class Definition is displayed (right top). The *Matching* button triggers the retrieval of matching sentences. The user can choose more positive and negative examples from the classified instances and iterate the procedure until the learned definition has an acceptable quality.

To aid the user during this process, the accuracy of the definition on the training

[11] available under `http://nachhalt.sfb632.uni-potsdam.de/owl`

data is given below the definition. Additionally, the number of matching sentences is displayed (in this case 5,299, ≈10% of the corpus). Hovering over a named class in the concept description presents a tooltip explaining the meaning of the construct as specified by the OLiA Annotation Model. This allows to quickly gain insight into the annotations of the corpus and judge whether the learning result matches the needs of the user.

4 Evaluation

In this section, we evaluate recall and precision of automatically acquired concepts for passive identification in German. We describe two problems (with 4 experiments each), in which we vary several configuration options: training set size (how many examples a user needs to choose), learning time and usage of lemmas.

4.1 Experimental Setup

We consider the German *werden* passive that is formed by the auxiliary *werden* and a past participle [23].

```
// root node
#root:[cat=/VP|S/] &
// root dominates "werden"
#root >* #werden:[lemma="werden"] &
// root dominates participle
#root >* #partizip:[pos=/V.PP/] &
// finite sentence
(#root >HD #werden |
// infinitive with zu
 (#root >HD #VZ:[cat="VZ"] & #VZ >HD #werden)) &
// root dominates participle directly
(#root >OC #partizip |
// participle is dominated by VP
 (#VP:[cat="VP"] >HD #partizip &
// root dominates VP directly
   (#root >OC #VP |
// or indirectly over a CVP
     (#root >OC #CVP:[cat="CVP"] & #CVP >CJ #VP))))
```

Figure 4: Rule for passive sentences in the Tiger Query Language [15]

The task is to distinguish passive clauses from other auxiliary constructions, given only linguistic surface structure (SSO) and morphosyntactic annotations (POS). In the corpus, neither POS nor SSO alone are sufficient to distinguish passive from active clauses, so that information from both sources has to be combined. For our experiment, the DL-Learner was trained on POS and lemmas. Syntax annotation was used only to identify target classifications (with the query in Fig. 4).
Three sets of sentences can be distinguished:

1. finite passive (finite auxiliary *werden*, 6,333 sentences, condition `#root >HD #werden`)
2. infinite passive with particle *zu* (lit. 'to') (37 sentences, condition `#root >HD #VZ`)
3. active (44,099 sentences that do not match the query)

From these sets we identified two learning problems to measure how well our approach can separate these sets from each another: (i) learn an OWL class that *covers* all finite passives (set 1) and the remainder (sets 2, 3), and (ii) distinguish between infinite passives (set 2) and the remainder. The second problem is especially difficult, as the number of correct sentences (37) is less than 0.07% of sentences in total.

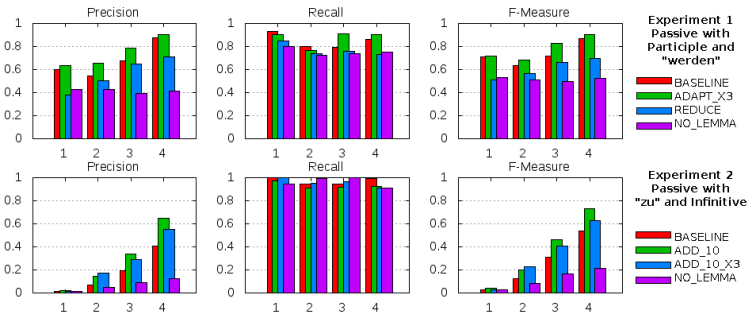

Figure 5: Evaluation results

For each problem, the data is split into training and test data (both positive and negative).[12]

As BASELINE, we randomly drew 5 positive (5p) and 5 negative (5n) sentences from the training data. In the experiments, we performed 4 iterations, starting with 5p+5n initial examples, and adding 5p+5n examples in every iteration. Precision and recall were measured on the intersection of retrieved sentences with the target classification.

We tested three configuration variations for the first problem: (1) we adapted the max. execution time to three times the number of examples (ADAPT, 30s, 60s, 90s, 120s),[13] (2) we reduced the number of initial examples to 2p+2n and added 2p+2n for each iteration (REDUCE, total 4,8,12,16), and (3) we deactivated the inclusion of *owl:hasValue* (lemmas) in the classes (NO_LEMMA).

As for the second problem, (1) we added 10 additional negative examples (ADD_10, total 20, 40, 60, 80), (2) we added 10n but adapted the runtime to 3 times the example size (ADD_10_X3, 60s, 120s, 180s, 240s), and (3) we used again the baseline (BASELINE) with no lemmas (NO_LEMMA).

For the first problem, we conducted a stratified leave-one-out 10-fold cross validation. As it was impossible to create 10 folds for the second set, we used a randomized 70%-30% split averaged over 10 runs (28 sentences for training, 11 for testing).

4.2 Results

Our results (summarized in Fig. 5) show that the Tiger Corpus Navigator is capable of acquiring concepts that involve multiple knowledge sources, here, the SSO (lemma) and the OLiA ontologies (for POS) with a high recall and with reasonable speed.

The observed high recall is inherent in the learning algorithm: When exploring

[12] Five overlapping sentences were removed.

[13] DL-Learner is an anytime algorithm, it stops when finding a class with 100% accuracy or a given maximum execution time (default 30 sec) is reached and returns its (intermediate) results.

the search spaces, it automatically discards all classes that do not cover all positive examples, so it produces very general results. High precision, however, can only be achieved after a certain number of iterations or by raising the *noise* parameter (zero in our experiments).

We found that our results are clearly dependent on lemmas, *owl:hasValue* inclusion yields better results. The selection of significant lemmas is done generically by DL-Learner according to a value frequency threshold, set equal to the number of positive examples. Users could also wish to manually configure this parameter or give certain lemmas in advance.

The size of the training set had a great influence on the performance with about 20% lower F-Measure in iteration 4 (REDUCE vs. ADD_10 to BASELINE). We observed marginal effects by increasing the maximum learning time with a slight F-Measure gain of 3.5% (ADAPT_X3 vs. BASELINE) and even a loss of more than 10% in the second experiment (ADD_10 vs. ADD_10_X3).

Although the second experiment amounts to a much lower F-Measure scores in iteration 4, the achieved results are interpretable: 40 % precision and 99 % recall mean that the retrieved set of sentences was reduced to about 100 sentences of which 40 would be correct. Such a small sample would be suitable for manual inspection and postprocessing.

Our implementation fulfills the speed requirements for a web scenario: For the first experiment, the average learning times for BASELINE were 1.8 sec, 22.6 sec, 31.9 sec and 29.5 sec, and for the second experiment 0.5 sec, 2.2 sec, 5.3 sec, 13.3 sec. The SPARQL queries needed 14.6 seconds on average and can be further improved by caching. The last example of the refinement chain in Sect. 2.1 was one of the highest scoring learned classes.

5 Related Work and Outlook

In the introduction, we identified three elementary functions a corpus tool has to fulfill, i.e., to **access**, to **document** and to **represent** linguistic annotations. We presented the Tiger Corpus Navigator, which provides *access* via a an intuitive user interface over the Web. The paradigm of navigating a corpus based on example sentences rids the necessity of being familiar with the *documentation* beforehand. Even more so, only the necessary information is presented unobstrusively on-the-fly. Learned classes *represent* the results in a formal, yet easily understandable way and the evaluation has shown that it is possible to extract the desired information without much time or effort.

5.1 Access to Linguistic Annotations

Linguistic corpora can be accessed by several corpus tools, e.g., GATE [9], TGrep2 [21], TigerSearch [15], the Stockholm TreeAligner [17], or MMAX2 [18], just to name a few. Newer tools also provide web interfaces, such as the IMS Corpus

Workbench [8], the Linguist's Search Engine [20], or ANNIS [7, 24].

All these tools, however, have in common that they operate on a formal, complex query language that represents a considerable hurdle to their application by non-specialists.

The Tiger Corpus Navigator represents an innovative approach to access corpus data that may complement such traditional corpus interfaces. It provides access to the primary data of specific sentences on the basis of extensionally defined conceptual descriptions, it is thus even possible to search for concepts that are not directly annotated (as shown for the passive concept and the Tiger POS annotation).

5.2 Document Linguistic Annotations

In our approach, linguistic annotations are explicitly documented by their linking to repositories of linguistic terminology. These repositories contain descriptions, definitions and examples that are represented to the user as tooltips (Fig. 3). In this way, the OWL representation of linguistic corpora and their linking with existing terminology repositories serves a documentation function.

And more than this, the application of the Tiger Corpus Navigator does not require the users to be familiar with the documentation at all: The automatic acquisition of query concepts allows a relatively uninformed user to run queries against a database without the necessity to be aware of the underlying data format, its expressivity and even the kind of annotations available. Thereby, our approach extends and generalizes approaches to access annotated corpora on the basis of abstract, ontology-based descriptions such as [19, 7]. As opposed to these, however, the concepts are not pre-defined in our scenario, but acquired by the system itself. The Tiger Corpus Navigator thus allows for corpus querying independently from the theoretical assumptions underlying the actual annotations in the corpus.

5.3 Represent Linguistic Annotations

As for exchange and representation formats, the linguistic community still struggles to define its own standards; several concurrent proposals are currently in use, e.g., NITE XML [4], UIMA XML [11], LAF/GrAF [14], or PAULA [7]. Here, standards from the Semantic Web community are applied, RDF and OWL, that are maintained by a large community and supported by a number of tools. So far, only few NLP tools working with OWL are available, e.g., [1], but a number of linguistic resources has already been transformed to OWL/DL [22, 3], or linked with ontologies [13]. Also, existing ontologies have been extended with concepts and properties for linguistic features [2, 10]. The Navigator represents another step in this development of convergence of ontological and NLP resources.

5.4 Application Scenarios

The Tiger Corpus Navigator may not constitute a full-fledged substitute for existing query tools, as the subsequent refinement of the classifier by the user may turn out to be a time-consuming task. It does, however, represent a prototype implementation of a technology that may be integrated with "traditional" tools to browse, query or access/distribute corpora. If used as a corpus exploration interface of an archive of linguistic resources, for example, the Tiger Corpus Navigator reduces the initial bias to assess the suitability of a corpus with unknown annotations. Such an archive may host different resources that require specialized tools for visualization and querying (e.g., TGrep2 for constituent syntax, MMAX2 for coreference, etc.), so that the efforts required to evaluate the suitability of a resource are enormous (a user has to acquaint itself not only with the annotations and some "standard" query language, but also with several specialized tools and their task-specific query languages). Using the Navigator, a user develops a classifier for a concept of interest, and the correctness of the classifier and the concept description obtained and the tooltips that contain their documentation allow her to assess the suitability of a corpus and its annotations for the task at hand immediately . If indeed a resource appears to be useful for a particular task, the user may decide to obtain the corpus and to process it further with the appropriate corpus tools.

5.5 Future Work

Future work includes the ability to save learned OWL classes. They can be collaboratively reused and extended by multiple users (Web2.0). Furthermore, they can be utilized to classify previously untagged text, converted by NLP2RDF in the same manner as here and thus extend the discovery of matching sentences beyond the initial corpora. With a corresponding parser-ontology pair it is even possible to replace the initial full text search by entering any example sentences.

It should be noted here that we aimed primarily for a proof-of-concept implementation. The Tiger Corpus Navigator does currently not come with an appropriate visualization, and it is restricted to sentence-level classification. Given sufficient interest from the community, the corresponding extensions may, however, be possible in subsequent research. Another topic for further research may be the combination of existing corpus management and corpus query tools with the Tiger Corpus Navigator, resp. the underlying technologies.

Acknowledgements

We would like to thank three anonymous reviewers for helpful comments and feedback. The work of Sebastian Hellmann, Jörg Unbehauen and Axel-Cyrille Ngonga Ngomo was supported by the Eurostars SCMS project, the work of Axel-Cyrille Ngonga Ngomo additionally by the Research Focus Media Convergence at the Johannes Gutenberg University of Mainz. The work of Christian Chiarcos was

supported by the German Research Foundation (DFG) in the context of Collaborative Research Center (SFB) 632 “Information Structure”, Project D1 “Linguistic Database”, University of Potsdam.

References

[1] G. Aguado de Cea, J. Puch, and J.Á. Ramos. Tagging Spanish texts: The problem of “se”. In *LREC 2008*, Marrakech, Morocco, May 2008.

[2] P. Buitelaar, T. Declerck, A. Frank, S. Racioppa, M. Kiesel, M. Sintek, R. Engel, M. Romanelli, D. Sonntag, B. Loos, V. Micelli, R. Porzel, and P. Cimiano. LingInfo: Design and applications of a model for the integration of linguistic information in ontologies. In *LREC 2006*, Genoa, Italy, May 2006.

[3] A. Burchardt, S. Padó, D. Spohr, A. Frank, and U. Heid. Formalising multilayer corpora in OWL/DL – Lexicon modelling, querying and consistency control. In *IJCNLP 2008*, Hyderabad, January 2008.

[4] J. Carletta, S. Evert, U. Heid, J. Kilgour, J. Robertson, and H. Voormann. The NITE XML Toolkit: flexible annotation for multi-modal language data. *Behavior Research Methods, Instruments, and Computers*, 35(3):353–363, 2003.

[5] C. Chiarcos. An ontology of linguistic annotations. *LDV Forum*, 23(1):1–16, 2008.

[6] C. Chiarcos. Grounding an ontology of linguistic annotations in the Data Category Registry. In *LREC-2010 Workshop on Language Resource and Language Technology Standards (LT&LTS)*, Valetta, Malta, May 2010.

[7] C. Chiarcos, S. Dipper, M. Götze, U. Leser, A. Lüdeling, J. Ritz, and M. Stede. A flexible framework for integrating annotations from different tools and tagsets. *TAL (Traitement automatique des langues)*, 49(2), 2008.

[8] O. Christ. A modular and flexible architecture for an integrated corpus query system. In *Papers in Computational Lexicography (COMPLEX ’94)*, pages 22–32, 1994.

[9] H. Cunningham. GATE, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254, 2002.

[10] B. Davis, S. Handschuh, A. Troussov, J. Judge, and M. Sogrin. Linguistically light lexical extensions for ontologies. In *LREC 2008*, Marrakech, Morocco, May 2008.

[11] T. Goetz and O. Suhre. Design and implementation of the UIMA Common Analysis System. *IBM Systems Journal*, 43(3):476–489, 2004.

[12] S. Hellmann. The Semantic Gap of Formalized Meaning. In *European Semantic Web Conference (ESWC)*, Heraklion, Greece, May 2010.

[13] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. Ontonotes: the 90% solution. In *HLT-NAACL 2006*, pages 57–60, New York, June 2006.

[14] N. Ide. GrAF: A graph-based format for linguistic annotations. In *ACL-2007 Linguistic Annotation Workshop*, Prague, Czech Republic, June 2007.

[15] E. König and W. Lezius. The TIGER language - a description language for syntax graphs, Formal definition. Technical report, IMS, 2003.

[16] J. Lehmann. DL-Learner: learning concepts in description logics. *Journal of Machine Learning Research*, 2009.

[17] T. Marek, J. Lundborg, and M. Volk. Extending the TIGER query language with universal quantification. In *Proceeding of KONVENS-2008*, pages 3–14, Berlin, October 2008.

[18] C. Müller and M. Strube. Multi-level annotation of linguistic data with MMAX2. In *Corpus Technology and Language Pedagogy*, pages 197–214. Peter Lang, Frankfurt am Main, 2006.

[19] G. Rehm, R. Eckart, and C. Chiarcos. An OWL-and XQuery-based mechanism for the retrieval of linguistic patterns from XML-corpora. In *RANLP 2007*, Borovets, Bulgaria, September 2007.

[20] P. Resnik, A. Elkiss, E. Lau, and H. Taylor. The web in theoretical linguistics research: Two case studies using the Linguist's Search Engine. In *31st Meeting of the Berkeley Linguistics Society*, pages 265–276, February 2005.

[21] D. Rohde. TGrep2 user manual, version 1.15. Technical report, MIT, Cambridge, MA, May 2005.

[22] J. Scheffczyk, A. Pease, and M. Ellsworth. Linking FrameNet to the Suggested Upper Merged Ontology. In *Fourth International Conference on Formal Ontology in Information Systems (FOIS 2006)*, pages 289–300, Baltimore, November 2006.

[23] G. Schoenthal. *Das Passiv in der deutschen Standardsprache*. Hueber, München, 1975.

[24] A. Zeldes, J. Ritz, A. Lüdeling, and C. Chiarcos. ANNIS: A search tool for multi-layer annotated corpora. In *Proceedings of Corpus Linguistics*, pages 20–23, Liverpool, UK, July 2009.