Ontology-based Entity Recognition and Annotation*

Thomas Hoppe[12], Jamal Al Qundus[1], Silvio Peikert[1]

[1] Fraunhofer-Institut FOKUS, Berlin, Germany
[2] Hochschule für Technik und Wirtschaft, Fachbereich 4, Angewandte Informatik, Berlin, Germany
{jamal.al.qundus, thomas.hoppe, silvio.peikert}@fokus.fraunhofer.de

Abstract. The majority of transmitted information consists of written text, either printed or electronically. Extraction of this information from digital resources requires the identification of important entities. While Named Entity Recognition (NER) is an important task for the extraction of factual information and the construction of knowledge graphs, other information such as terminological concepts and relations between entities are of similar importance in the context of knowledge engineering, knowledge base enhancement and semantic search. While the majority of approaches focusses on NER recognition in the context of the World-Wide-Web and thus needs to cover the broad range of common knowledge, we focus in the present work on the recognition of entities in highly specialized domains and describe our approach to ontology-based entity recognition and annotation (OER). Our approach, implemented as a first prototype, outperforms existing approaches in precision of extracted entities, especially in the recognition of compound terms such as *German Federal Ministry of Education and Research* and inflected terms.

Keywords: Ontology, Entity Recognition, Text Annotation, DBpedia Spotlight, BioPortal Annotator.

1 Introduction

Two realms define the range in which entity recognition has to take place. One realm needs to cover a large and broad range of common entities, related to common knowledge and contained in the broad range of web resources and documents, largely consisting of factual information about named entities. The other one covers highly specialized information in monothematic application domains and has a strong focus on the terminology used in the domain. Although it often covers also a large, but still limited set of entities, these entities are usually identified by complex names, such as

* This work has been partially supported by the "Wachstumskern Qurator – Corporate Smart Insights" project (03WKDA1F) funded by the German Federal Ministry of Education and Research (BMBF).

chemical compounds Acetylsalicylic Acid, job titles Servicetechniker für Windkraftanlagen, etc.

Recognition of entities in the first realm is known under the term *named entity recognition* (NER). Since these approaches are often based on large common corpora, they are usually generic and domain-independent, but applicable to a broad range of application areas. Although they can cope with text in arbitrary domains, these approaches have problems recognizing all important entities in an application domain. Because of this incompleteness, they can achieve only limited recall. Further, their precision is limited by missing information.

We summarize approaches of the second realm, which are not limited to named entities, under the more general term *entity recognition* (ER). These approaches rely on given background knowledge about the entities in a particular domain. Thus, they are domain-dependent and applicable to a smaller range of domains, but configurable by the background knowledge. Their goal is to detect as much relevant domain entities as possible, enabling thus higher recall and precision. If such an approach uses knowledge formalized as ontology[1], we term it *ontology-based entity recognition* (OER).

An example of NER in the first realm is – besides others – DBpedia Spotlight, an open source annotation tool for recognizing named entities in text and linking them to DBpedia resources [2]. DBpedia Spotlight can be trained with Wikipedia content for different languages. The quality of its entity spotting approach thus depends on the documents available in a particular language-dependent Wikipedia. Connected with this approach are additional limitations. Inflected and compound terms are often not recognized as [3] point out. The coverage of entities from specialized domains is uneven: while e.g. VIPs and Genomics will be covered in depth, products of a particular company will be just covered on the surface.

Approaches of the first realm often require disambiguation mechanisms to decide which interpretation of a named entity is in a certain context intended. DBpedia Spotlight either tries to identify the right interpretation automatically or offers the user to select one of the interpretations [2].

Entity recognition in the second realm is based on the availability of given controlled vocabularies, which may originate from lists of terms, taxonomies, thesauri up to ontologies. Although several papers describe and evaluate their systems, they are usually not publically available. One exception here is the BioPortal Annotator[2], which performs ER and annotation of documents based on a larger number of biomedical ontologies. But the BioPortal Annotator has problems too: if more than one ontology is used for annotation its results are highly redundant. Its ability to recognize compound terms is limited and, even if just one ontology is used for ER, it is rather slow.

Disambiguation plays a minor role in this second realm, since the number of polysemic terms in controlled vocabularies is usually rather small. Although a term, like

---

[1] We use the term ontology in the sense of [1] as "... an explicit, shared specification of a conceptualization" and interchangeable with knowledge model.

[2] https://bioportal.bioontology.org/annotator (last access Dec., 13th 2019)

construction, can be used in a narrow domain with different meanings, as *process*, *department*, or *task*, these meanings are often strongly related. Hence, a clean differentiation of these meanings is not always necessary.

For complementing DBpedia Spotlight, we decided to follow a similar approach as [3,4] and develop a fast Ontology-based Recognition and Annotation system (OER), which accounts for common spelling errors, inflected and compound terms. In contrast to [3] however, we base the system on controlled vocabularies obtained from knowledge models. Although our approach is based like [3] on a two-layered transducer architecture for the recognition process, we simplified the recognition process. [3] uses a parallel multi-process approach looking ahead for compound terms in order to avoid backtracking. This approach may deliver several alternative compounds; therefore, a voting process chooses the best compound, i.e. the longest matching compound. We use a single process instead, which scans through the text looking directly recursively ahead for the longest compounds. This approach avoids backtracking too, by deciding which longest sequence to keep when ascending from the recursive lookahead.

2 Architecture

The architecture of OER consists of two parts working in two subsequent phases: during the first compilation phase, a language-dependent lemmaCache is initialized and the terminology of a knowledge model is precompiled into a lookupDictionary. The second annotation phase uses the precompiled lemmaCache and the lookupDictionary for the annotation of texts as shown in **Fig. 1**.

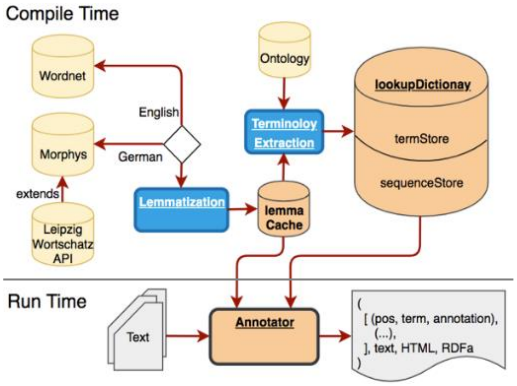

Fig. 1. OER Architecture

2.1 Lemmatization 词形还原

For the initialization of the lemmaCache different sources are consulted depending on the language. For German the lemmaCache is initialized on the base of a dump of Morphys morphology dictionary[3], which allows lemmatizing more than 400.000 German word forms directly from the start. This cache is augmented during runtime via the API of Wortschatz Leipzig with additional lemmas derived from 1.000.000 sentences from Wikipedia or a news corpus. For English the lemmas are derived currently from Wordnet only.

[3] http://www.danielnaber.de/morphologie/ (last access Dec., 13th 2019)

2.2 Terminology Extraction

A knowledge model of a domain is used as source for the derivation of a controlled vocabulary consisting of (term,URI)-tuples. The terms, which may consist of single tokens or token sequences, form the controlled vocabulary for the entity recognition. The URIs build the values used for annotating the recognized entities.

By default the terms are derived for knowledge models in OWL, RDFS and SKOS from rdfs:label, skos:predLabel, skos:altLabel or skos:hiddenLabel. However, since a knowledge model may consist of more complex structures built from concepts, preferred terms and their preferred labels, we also allow users of OER to define their own derivation pattern for (term,URI)-tuples via a user-defined SPARQL call. For simpler knowledge models, we also allow the specification of (term,URI)-tuple via CSV.

Especially, for German it is important that the entity recognition can recognize different spelling variants of the same entity. German is famous for its compound nouns, creating new nouns by connecting adverbs (Soforthilfe), adjectives (Dreirad), verbs (Fahrlehrer) and nouns (Mädchenhandelsschule). However, under certain circumstances parts of a compound can be separated by a hyphen in order to improve legibility and to avoid ambiguity (Mädchen-Handelsschule). As experience has shown during the analysis of search queries, authors and users often even separate these parts incorrectly by blanks (Robert Koch Institut)4. These deficiencies may also occur combined in different variations, e.g. (Johannes Gutenberg-Universität Mainz). In order to recognize all these variations easily as referring to the same entity or concept, we compute them in advance and store them in a two-layered prefix tree structure.

2.3 Two-Layered Tree-based Recognizer

The first layer is based on a radix tree that builds a termStore. Each lemmatized term contained in a term sequence is used as key of the termStore to store and access a unique id for each lemma. The list of unique ids of each term sequence are used subsequently as key in a prefix tree called sequenceStore forming the second layer. These lists of unique ids are used to store and access URIs of entities and concepts corresponding to term sequences.

Thus, a term sequence like *gewählter Abgeordneter des deutschen Bundestags*, will be lemmatized and normalized as *gewählt abgeordnete des deutsch bundestag* which in turn is translated into a numerical list, e.g. [2643, 92, 83634, 12344. This encoding of the term sequence is used as list-based key to access the URI of the corresponding knowledge model concept in the sequenceStore. This translation safes space through the numerical encoding of strings and allows mapping different flexions of labels, such as *gewählten Abgeordneten des deutschen Bundestag*, to the same URI.

---

4 https://deppenleerzeichen.de/ (last access Dec., 13th 2019)

2.4 Recognition and Annotation by Compound Term Lookahead

This two-layered data structure is initially set up during the compilation phase and used during run-time to scan a given text in order to recognize and annotate compound terms. Suppose that the knowledge model contains two additional concepts with the labels *gewählte Abgeordnete* and *deutscher Bundestag*. Assume further that we like to annotate the following text: *Als gewählte Abgeordnete des deutschen Fischzüchterverbandes reisen sie nach Berlin und treffen die gewählten Abgeordnete des deutschen Bundestags*.

The recognition and annotation process simply scans the tokenized text from the beginning until a term contained in the termStore is reached (see **Fig. 2**). In the example, this is *gewählte*. Starting from this term a lookahead is performed searching for the longest sequence of terms, which are contained in the termStore and which form a term sequence contained in the sequenceStore. As soon as a subsequent term is not included in the termStore, the lookahead process terminates and delivers the longest termSequence still contained in the termStore together with its length (see **Fig. 3**).

```
annotate (text, lc, ld):
  /* lc (lemmaCache), ld (lookupDictionary) */
  tt := tokenize(text); s := p := 0;  at := '';  a := []
  for token in tt:
    p := p + 1
    if s > 1:
      s := s - 1
      continue
    elseif not lc.lemmatize(token) in ld.termStore:
      at := at + ' ' + token
    else:
      (phrase,l) := lookahead(tt[p+1:],[term],1)
      if not phrase == []:
        URI = ld.get(phrase)
        if not URI == '':
          a := a.append((p,phrase,URI))
          at := at + ' ' + wrapHTML(phrase)
          continue
      at := at + ' ' + token
  return (a,at)
```

Fig. 2. Pseudo Code of Annotation Process

The lookahead identifies *gewählte Abgeordnete des deutschen* as a sequence of terms each included in the termStore. As soon as it determines that *Fischzüchterverbandes* is not included in the termStore, it will resort to the longest-term sequence found so far: *gewählte Abgeordnete*, since longer term sequences are not contained in the sequence store.

The longest-term sequences found are used to derive from the sequenceStore the corresponding annotation values, the length of the identified term sequences are used to skip the next n tokens in the scan pipeline.

Eventually, this process recognizes for the example text the annotations of the term sequences *gewählte Abgeordnete* and *gewählte Abgeordnete des deutschen Bundestags*.

```
lookahead(tt, fp ,n):
  if tt == []:
    return (n,fp)
  termfound := lc.lemmatize(tt[0]) in ld.termStore
  phraseFound := fp in ld.phraseStore
  if termFound or phraseFound:
    (ph,l) := lookahead(tt[1:],fp.append(tt[0]),n+1)
    if ph in ld.phraseStore:
      return (ph,l)
    elseif phraseFound:
      return (fp,n)
  return ([],n)
```

Fig. 3. Pseudo Code of LookAhead Procedure

3 Evaluation

In a first evaluation during the development, we compared this solution with DBpedia Spotlight on recruitment related German texts and with the BioPortal Annotator on medical texts in English using the MeSH ontology. In both cases, our system is able to identify compound terms in German as well as in English.

3.1 Recruitment Domain

For a first evaluation of OER's annotations against DBpedia Spotlight, we used the Recruitment Thesaurus of Ontonym[5] currently consisting of more than 16.000 concepts and more than 20.000 labels – partially multilingual. As illustration, the following text excerpt from Wikipedia leads to the annotations shown in **Fig. 4**:

„*Medizinisch-technischer Assistent (MTA) ist die Sammelbezeichnung für die vier Berufsbilder der technischen Assistenten in der Medizin und Tiermedizin im deutschen Gesundheitswesen. Sie umfasst im Einzelnen die Ausbildungsberufe:*

- *Medizinisch-technischer Assistent – Funktionsdiagnostik (MTAF)*
- *Medizinisch-technischer Laboratoriumsassistent (MTLA oder MTA-L)*
- *Medizinisch-technischer Radiologieassistent (MTRA, MTA-R oder RTA)*

[5] A former spin-off (2008 - 2015) from the Freie Universität Berlin and the first author.

- *Veterinärmedizinisch-technischer Assistent (VMTA)*

Der Namensbestandteil „-assistent“ kann zur Verwechslung mit dem Beruf des medizinischen Fachangestellten (Arzthelfer) führen, der sich in Ausbildung und Tätigkeit aber deutlich unterscheidet.“

Fig. 4. Comparison of Annotations from DBpedia Spotlight (top) and OER (bottom)

Fig. 4 and **Table 2** (in the appendix) indicate that with the use of specialized domain knowledge the recall and precision of annotation can be improved in comparison to an annotator using a broader and more general knowledge model.

Medizinisch-technischer Assistent (MTA) ist die Sammelbezeichnung für die vier Berufsbilder der technischen Assistenten in der Medizin und Tiermedizin im deutschen Gesundheitswesen. Sie umfasst im Einzelnen die Ausbildungsberufe: Medizinisch-technischer Assistent – Funktionsdiagnostik (MTAF) Medizinisch-technischer Laboratoriumsassistent (MTLA oder MTA-L) Medizinisch-technischer Radiologieassistent (MTRA, MTA-R oder RTA) Veterinärmedizinisch-technischer Assistent (VMTA) Der Namensbestandteil „-assistent“ kann zur Verwechslung mit dem Beruf des medizinischen Fachangestellten (Arzthelfer) führen, der sich in Ausbildung und Tätigkeit aber deutlich unterscheidet.

3.2 Medical Domain

In a second evaluation, we compared OER with the BioPortal Annotator on medical texts annotated with MeSH[6] [4]. **Table 1** shows the annotations of the following text excerpt from Wikipedia:

"*Aspirin, also known as (Acetylsalicylic Acid), (ASA), is a medication used to treat pain, fever, or inflammation. Specific inflammatory conditions which aspirin is used to treat include Kawasaki disease, pericarditis, and rheumatic fever. Aspirin given shortly after a heart attack decreases the risk of death. Aspirin is also used long-term to help prevent further heart attacks, ischaemic strokes, and blood clots in people at high risk. It may also decrease the risk of certain types of cancer, particularly colorectal cancer. For pain or fever, effects typically begin within 30 minutes. Aspirin is a non-steroidal anti-inflammatory drug (NSAID) and works similarly to other NSAIDs but also suppresses the normal functioning of platelets.*"

In contrast to the mgrep based approach of the BioPortal Annotator, as identified in [5], OER is not only able to find compound terms of MeSH concepts, it even finds annotations of terms the BioPortal Annotator is not able to recognize.

Table 1. Annotations of the sample text[8]

BioPortal Annotator	OER
aspirin : 5	aspirin : 5
risk : 3	risk : 3
pain : 2	pain : 2
fever : 2	fever : 2
	acetylsalicylic acid : 1
inflammation : 1	inflammation : 1
disease : 1	kawasaki disease : 1
pericarditis : 1	pericarditis : 1
rheumatic fever : 1	rheumatic fever : 1
heart : 2	heart attack : 1
	heart attacks : 1[7]
death : 1	death : 1
	strokes : 1
blood : 2	blood clots : 1
	cancer : 1
	colorectal cancer : 1
	nsaids : 1
	platelets : 1

4 Summary

Entity Recognition is an important task for the identification of information in written text. To address this challenge, we have implemented a first prototype of an Ontology-based Recognition and Annotation system (OER) which is fast and can handle common spelling mistakes, flections, and compound terms. The architecture of OER supports two phases. In a first compilation phase, a language-dependent lemmaCache is initialized and a knowledge model is precompiled into a lookupDictionary, allowing to identify terms of the controlled vocabulary quickly and to retrieve their corresponding concept URIs. The second annotation phase uses these data structures to annotate texts by

[6] https://www.nlm.nih.gov/mesh/meshhome.html (last access Dec., 13th 2019)
[7] This difference is caused by WordNets inability to lemmatize "attacks".
[8] Numbers indicate the number of occurrences of each term.

a single-threaded recursive scanning process of the text, delivering always the longest matching term sequence. We could show that OER gives, through the usage of domain knowledge, better annotations than DBpedia Spotlight. In contrast to the BioPortal Annotator, its annotations are more complete and it identifies compound terms better.

Currently OER is still in a prototype phase and has some limitations. One of these limitations is the lemmatization of German compound terms. Since such compounds usually do not appear in morphologic dictionaries, we intend to augment the lemmaCache by a simple approach for splitting compounds, lemmatizing their head term and joining the lemmatized fragments together. Another limitation is the treatment of the different notations of gender-neutral terms, which can be solved rather easily. Because of the nature of the texts and domains we like to process with the system, we deliberately ignored the question of disambiguation for the initial development.

Of course, one limitation slips in by the used knowledge models: only the terms contained in the knowledge model, their lemma and word forms related to these lemmas can be recognized by this approach. Therefore, the annotations will only be as good as the knowledge models themselves. However, we do not regard this as a limitation; instead, we consider it a feature, since it allows focusing on entities contained in the knowledge model of a target domain [6].

Besides the lemmatization of compounds and the treatment of gender-neutral terms, an interesting, more experimental augmentation of the system would be the recognition of the semantic equivalence of certain noun phrases and compound nouns. Additionally the annotation process could be extended by annotating terms with categorical information and limiting the number of annotated terms based on a numerical measure of their specificity. Of course, further code optimizations and investigations of the quality of the annotations as well as of the speed of the annotation process need still to follow.

References

[1] R. Studer, V.R. Benjamins, D. Fensel, ‘Knowledge Engineering: Principles and Methods’. *Data and Knowledge Engineering* 25(1-2):161-197, Elsevier, 1998.

[2] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer, ‘DBpedia spotlight: shedding light on the web of documents’, in *Proceedings of the 7th international conference on semantic systems*, 2011, pp. 1–8.

[3] C. Jilek, M. Schröder, R. Novik, S. Schwarz, H. Maus, and A. Dengel, ‘Inflection-tolerant ontology-based named entity recognition for real-time applications’, *ArXiv Prepr. ArXiv18120.2119*, 2018.

[4] C. Jonquet, N. Shah, M. Musen, ‘A System for Ontology-Based Annotation of Biomedical Data’, In: A. Bairoch, S. Cohen-Boulakia, C. Froidevaux (eds) *Data Integration in the Life Sciences*. DILS 2008. Lecture Notes in Computer Science, Vol 5109. Springer, Berlin, Heidelberg.

[5] D. Sanchez-Cisneros, F. Aparicio Gali, ‘An Ontology-based namedentity recognition system for biomedical texts’, *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation* (SemEval 2013).

[6] Miha Štravs, Jernej Zupančič, ‘Named Entity Recognition Using Gazetteer of Hierarchical Entities’, 10.1007/978-3-030-22999-3_65, in: *Advances and Trends in Artificial Intelligence. From Theory to Practice*, F. Wotawa, et al. (Eds.), LNAI 11606, pp. 768-776, Springer Nature, 20195

5 Appendix

Term OER	**Annotation OER**	**Term DBpedia Spotlight**	**Annotation DBpedia Spotlight**
Medizinisch-technischer Assistent	ont:Medizinisch-Technischer-Assistent	Medizinisch-technischer Assistent	dbp:Medizinisch-technischer_Assistent
MTA	ont:Medizinisch-Technischer-Assistent	MTA	dbp:Medizinisch-technischer_Assistent
Berufsbilder	ont:Beruf		
technischen Assistenten	ont:technischer_Assistent	technischen Assistenten	dbp:Technischer_Assistent
Medizin	ont:Medizin	Medizin	dbp:Medizin
Tiermedizin	ont:Tiermedizin	Tiermedizin	dbp:Veterinärmedizin
deutschen	ont:Deutsch		
Gesundheitswesen	ont:Gesundheitswesen	Gesundheitswesen	dbp:Gesundheitssystem
Ausbildungsberufe	ont:Ausbildungsberuf		
Medizinisch-technischer Assistent	ont:Medizinisch-Technischer-Assistent	Medizinisch-technischer Assistent	dbp:Medizinisch-technischer_Assistent
Funktionsdiagnostik	ont:Funktionsdiagnostik	Funktionsdiagnostik	dbp:Medizinische_Untersuchung
MTAF	ont:Medizinisch-Technischer-Assistent_fuer_Funktionsdiagonstik		
Medizinisch-technischer Laboratoriumsassistent	ont:Medizinisch-Technischer_Laboratoriumsassistent	Medizinisch-technischer Assistent	dbp:Medizinisch-technischer_Assistent
MTLA	ont:Medizinisch-Technischer_Laboratoriumsassistent		
MTA-L	ont:Medizinisch-Technischer_Laboratoriumsassistent		
		Radiologieassistent	dbp:Radiologie
Medizinisch-technischer Radiologieassistent	ont:Medizinisch-Technischer_Radiologieassistent		
MTRA	ont:Medizinisch-Technischer_Radiologieassistent	MTRA	dbp:Medizinisch-technischer_Assistent
		RTA	dbp:Radio_Television_Afghanistan
MTA-R	ont:Medizinisch-Technischer_Radiologieassistent		
Veterinärmedizinisch-technischer Assistent	ont:Veterinaermedizinisch-technischer_Assistent		
VMTA	ont:Veterinaermedizinisch-technischer_Assistent		
Beruf	ont:Beruf		
medizinischen Fachangestellten	ont:medizinische_Fachangestellter	medizinischen Fachangestellten	dbp:Medizinischer_Fachangestellter
Arzthelfer	ont:Arztfachhelfer	Arzthelfer	dbp:Medizinischer_Fachangestellter
führen	ont:Leitung		
Ausbildung	ont:Ausbildung		
Tätigkeit	ont:Aufgabe		

Table 2. Comparison of OER and DBpedia Spotlight Annotation[9]

[9] Name spaces of URIs are abbreviated. Terms and annotations found by either OER or DBpedia Spotlight alone are marked in green. Red marks wrong annotations and orange marks annotations, which are correct but not precise.