AGDISTIS - Agnostic Disambiguation of Named Entities Using Linked Open Data

Ricardo Usbeck[1 2 ⋆], Axel-Cyrille Ngonga Ngomo[1], Sören Auer[1], Daniel Gerber[1], and Andreas Both[2]

[1] University of Leipzig, Germany
{usbeck|ngonga|auer|dgerber}@informatik.uni-leipzig.de
[2] R & D, Unister GmbH, Leipzig, Germany

Abstract. One of the key steps towards extracting RDF from natural-language corpora is the disambiguation of named entities. While several approaches aim to address this problem, they still achieve poor accuracy on Web data. In this paper, we present AGDISTIS, a novel named entity disambiguation approach that performs well on Web data. Our approach combines an extension of the HITS algorithm with label expansion and string similarity measures. Based on this combination, it can efficiently detect the correct URIs for a given set of named entities within an input text. AGDISTIS is agnostic of the underlying knowledge base and can thus be used on any RDF knowledge base and for any language. We evaluate our approach on three different news datasets against a state-of-the-art named entity disambiguation framework. Our results indicate that we outperform the state-of-the-art approach by up to 20% accuracy. We also measure the effect of the structure of the underlying knowledge base on the accuracy of our system. Here, our results suggest that only a few RDF properties contribute to boosting the accuracy of the disambiguation.

1 Introduction

Realizing the vision of a usable and up-to-date Linked Data Web requires approaches that allow extracting RDF from data sources of different types, i.e., unstructured, semi-structured and structured data sources in real time. While several approaches have been designed to deal with such data sources, the accurate processing of unstructured data from the Web remains a tedious endeavor. Yet, current approaches for extracting RDF data from unstructured Web data streams suffer from two main drawbacks: First, current approaches perform poorly on Web text [17]. This is due to Web text being more noisy than other text sources as well as containing a large number of resources from very different domains. Second, most state-of-the-art approach rely on algorithms with non-polynomial time complexity [11,10] or exhaustive data mining methods [4,18]. Despite the recent development of approaches which harness Linked

⋆ We thank Michael Röder, Maximilian Speicher and Didier Cherix.

Data for entity recognition and disambiguation [10,9,15], there is still a huge potential for employing Linked Data background knowledge in various Natural Language Processing (NLP) tasks. With Linked Data and live RDF streams such as DBpedia Live[3], we have comprehensive and evolving background knowledge comprising information on the relationship between a large number of real-world entities. Consequently, we can exploit the graph structure and semantics of such background knowledge for deciding to what concept a certain string should be mapped. As a result, the accuracy of natural language processing services can be boosted by improving and adding more background knowledge sources. Ultimately, this might give rise to a new paradigm of semantics-based NLP services, which truly leverage the rich semantics of community-generated, multilingual and evolving LOD background knowledge.

Example 1 *Barack Obama arrived this afternoon in Washington, D.C.. President Obama's wife Michelle accompanied him.*

The extraction of RDF from unstructured sources comprises three main steps: named entity recognition, entity disambiguation and relation extraction. *Named entity recognition* (NER) aims to discover strings which stand for labels of entities which instantiate predefined classes, most commonly organizations, persons or locations. These instances are called *named entities*. For example, for the first sentence of Example 1, an accurate entity recognition approach would return the strings `Barack Obama` and `Washington, D.C.` *Named entity disambiguation* (NED) uses a given input text and already recognized named entities to assign and map each entity to a canonical form, e.g., the URI of a corresponding Linked Data resource. For example, the strings `Barack Obama` and `Washington, D.C.` can be mapped to the resources `dbr:Barack_Obama` and `dbr:Washington,_D.C.`[4] respectively, originating from DBpedia [2]. *Relation extraction* (RE) builds upon NER and NED to extract possible connections between entities using unsupervised methods based on Wikipedia [21]. In this paper, we focus exclusively on the entity disambiguation step and assume that the entity recognition is given.

We address the problems of accuracy and scalability by presenting *AGDISTIS*, a NED approach and framework. Our approach combines the HITS [12] algorithm with label expansion and string similarity measures. HITS is a fast graph algorithm which runs with an upper bound of $\Theta(k \cdot x)$ with k the number of iterations and x the number of nodes in the graph. Since the graph algorithm can be computed document-wise, our approach can be easily implemented in parallel and deployed at web scale. Furthermore, our results suggest that we outperform the state of the art by up to 20% accuracy. Our contributions can be summed up as follows:

- We present AGDISTIS, a framework and approach for disambiguating named entities that is agnostic to the underlying knowledge base (KB).
- In particular, we introduce a novel real-time NED algorithm based on a combination of the HITS algorithm and of linguistic heuristics.

[3] `http://live.dbpedia.org`
[4] `dbr` stands for http://dbpedia.org/resource/

– We evaluate AGDISTIS on three multilingual, multi-domain and open-source datasets extracted from the Web.[5]
– We show that our approach outperforms the state of the art on these datasets.
– We evaluate the robustness of our approach against the topology of the underlying KB by measuring the influence of the properties in the KB on the accuracy of AGDISTIS.

The remainder of this paper is organized as follows: We first give a brief overview of related work. Then, we formalize the task of NED in Section 3. The AGDISTIS approach is presented in Section 4 comprising an overview, the way AGDISTIS detects candidates and the disambiguation algorithm itself. After presenting the datasets, we evaluate our approach against AIDA [9] and measure the influence of using surface forms in Section 5. Thereafter, we analyze the contribution of certain properties to our disambiguation approach in Section 6, concluding in Section 7 highlighting emerging research questions.

2 Related Work

AGDISTIS is related to the research area of Information Extraction (IE) [14] in general and to NED in particular. Several approaches have been developed to this end. For example, *Epiphany* [1] identifies, disambiguates and annotates entities in a given HTML page with RDFa. Epiphany can be deployed on arbitrary Linked Data using FOAF[6] as open vocabulary for annotation. Ratinov et al. [18] recently described an approach for disambiguating entities to Wikipedia. The authors argue that using Wikipedia can lead to better global approaches than using traditional local algorithms. Cucerzan presents an approach based on extracted Wikipedia data towards disambiguation of named entities [4]. The author tried to maximize the agreement between contextual information of Wikipedia pages and the input text, i.e., local approach. Furthermore, the system considered the category tags associated with candidate entities.

Kleb et al. [11,10] developed and improved an approach using ontologies to mainly identify geographical entities but also people and organizations in an extended version. These approaches comprised a combination of DBpedia and Geonames[7] as the underlying knowledge base. They evaluated their approach using two subsets of the Reuters-21578[8] corpus which were manually annotated. Results are promising showing a precision of up to 100% while the recall is limited to 90.9% for that simple task. In an extension of their algorithm they elevate their approach by using spread activation and Steiner-Trees on RDF-based graphs [10]. This global approach leads to an f-measure of 64.68% using a 46 document comprising corpus extracted from the European Media Monitor [16].

[5] The datasets as well further data and source code for this paper can be found at `http://github.com/AKSW/AGDISTIS`.
[6] `http://www.foaf-project.org/`
[7] `http://www.geonames.org/`
[8] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

Moreover, Kleb et al. present three extensions comprising text windows for local contexts, adding bidirectional explorations during spread activation phase and applying reinforcement learning [11]. Results from 164 out of 200 news from [16] show an f-measure of 75%. The disadvantage of this approach is the use of Steiner Trees, which is NP-complete, and spread activation algorithms w.r.t. to web scale document collections.

Another related research field is cross-document coreferencing, i.e., grouping entities in a corpus of documents. This problem has been tackled recently by Singh et al. [19] by using a parallel MCMC-based inference mechanism and a hierarchical coreference model. The authors demonstrate the web scalability of their approach by using a corpus comprising 1.5 million named entities.

The state-of-the-art algorithm *AIDA* [9] for named entity disambiguation is based on the YAGO[9] knowledge base and sophisticated graph algorithms. Specifically, this approach uses dense sub-graphs to identify coherent mentions using a greedy algorithm enabling web scalability. Additionally, AIDA disambiguates w.r.t. similarity of contexts, prominence of entities and context windows. In Section 5.2 we compare our approach to AIDA.

3 Named Entity Disambiguation

The goal of AGDISTIS is to detect correct resources for n a-priori determined named entities N in a natural language text. We will introduce a formalization of this problem here. In general, several resources from a given knowledge base K share enough commonalities with a named entity to be considered as candidate resources for the entity. For the sake of simplicity and without loss of generality, we will assume that each of the entities can be mapped to m resources. We call the j^{th} candidate for the i^{th} named entity C_{ij} and denote the matrix which contains all candidate-entity mappings by C. Let μ be a family of functions which maps each entity N_i to exactly one candidate C_{ij}. We call such functions *assignments*. The output of an assignment is a candidate vector of length $|N|$ that is such that the i^{th} entry of the vector maps with the i^{th} named entity from N. Let ψ be a function which computes the similarity between an assignment of candidates from the knowledge base and the vector of named entities N. The *coherence* function ϕ models the similarity of the input text and a vector of possible mapping resources each of the candidates (cf. [18]). Given this formal model, the goal of AGDISTIS is to find the assignment $\mu^\star$ with

$$\mu^\star = \arg\max_{\mu} \sum_{i} \left(\psi(\mu(C,N),N_i) + \phi(\mu(C,N),K)\right).$$

This problem has obviously a non-polynomial time complexity. Thus, for the sake of scalability, AGDISTIS computes an approximation of $\mu^\star$ by solving an easier problem.

[9] `http://www.mpi-inf.mpg.de/yago-naga/yago/`

4 The AGDISTIS Approach

In the following, we present the AGDISTIS approach in detail. We begin by giving an overview of the approach. Then, we explain how to choose candidates to construct the candidate-entity mappings matrix C. Finally, we show how to approximate the assignments $\mu^\star$ efficiently.

4.1 Overview

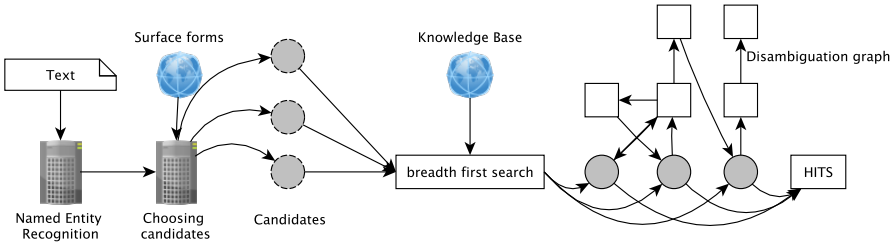

Fig. 1. Overview of AGDISTIS.

Our approach for named entity disambiguation consists of three main phases as depicted in Figure 1. Given an input text T and a named entity recognition function (e.g., [15,6]), we begin by retrieving all named entities from the input text. Thereafter, we aim to detect candidates for each of the detected named entities. To this end, we apply several heuristics and make use of known surface forms for resources from the underlying KB [13]. The set of candidates generated by the first step is used to generate a semantic context graph. Here, we rely on a graph search algorithm which retrieves context information from the underlying KB. Finally, we employ the HITS algorithm [12] to the context graph to find authoritative candidates for the discovered named entities.

4.2 Candidate Detection

In order to find the correct disambiguation for a certain set of named entities, we first need to detect candidate resources in the knowledge base. We begin by creating an index comprising all labels of each resource from the knowledge base. While our approach can be configured to use any set of properties as labeling properties, we rely on the properties for labeling resources presented in [5] as default. In addition, our approach can make use of known *surface forms* for each of the resources in case such knowledge is available [13]. These are simply added to the set of available labels for each resource, cf. Section 5.1.

Next to searching the index we apply a number of linguistic heuristics to account for natural-language deficiencies:

– An expansion policy accounting for coreferencing resolution.
– A string normalization based on eliminating plural and genitive forms, removing common affixes such as postfixes for enterprise labels and ignoring candidates with time information (years, dates, etc.) within their label.

The resulting candidate detection approach is explicated in Algorithm 1.

Algorithm 1: Searching candidate nodes for a specific label.

```
Data: label of a certain named entity, σ trigram similarity threshold
Result: boolean whether candidate nodes were found, C candidates found
begin
    addedCandidates ⟵ false, C ⟵ ∅;
    label ⟵ tryRemoveFormOfEnterprise(label);
    C̄ ⟵ searchIndex(label);
    if |C̄| == 0 then
        label ⟵ tryRemovePluralS(label);
        label ⟵ tryRemoveGenitiveS(label);
        C̄ ⟵ searchIndex(label);
    for c ∈ C̄ do
        if c.startsWith("http://dbpedia.org/resource/") then
            if ¬c .matches([0-9]+) then
                if ¬isDisambiguationSite(c) then
                    continue;
                if trigramSimilarity(c, label)< σ then
                    continue;
                c ⟵ redirect(c);
                if fitDomain(c) then
                    C ⟵ C∪ c;
                    addedCandidates ⟵ true;
```

Coreference resolution plays a central role when dealing with text from the Web. Especially, named entities are commonly mentioned in their full length the first time they appear within a document, while the subsequent mentions only consist of a substring of the original mention. For example, a text mentioning Barack Obama's arrival in Washington D.C. would use the strings `Obama` or `Barack` as labels for Barack Obama later in the same text (see Example 1). This is simply due to humans readers being able to carry out a co-reference analysis on the fly. AGDISTIS thus begins its disambiguation by employing the string expansion policy described in Algorithm 2. Our policy stores all named entity strings in order of their string length. If we recognize an entity string matching a part of an already processed entity, we expand the current string to the one stored earlier. This assumes both named entities mention the same instance. If

there are several possible expansions, we choose the shortest as an empirically good rule of thumb.

Algorithm 2: Search candidates for named entities and expansion policy

```
Data: N = {N_1, N_2 ... N_n} sorted in ascending order of their string length,
      trigram similarity threshold σ
Result: set of candidates C
begin
    heuristicExpansion ⟵ ∅, C ⟵ ∅ ;
    for N_i ∈ N do
        label ⟵ string(N_i);
        tmp ⟵ label;
        expansion ⟵ false;
        for key ∈ heuristicExpansion do
            if key contains label then
                if tmp.length > key.length && tmp != label then
                    tmp ⟵ key;
                    expansion ⟵ true;
                if tmp.length < key.length && tmp == label then
                    tmp ⟵ key;
                    expansion ⟵ true;
        label ⟵ tmp;
        if ¬expansion then
            heuristicExpansion ⟵ label ∪ heuristicExpansion
        C ⟵ C∪ searchCandidatesForALabel(label,σ);
```

After expanding named entities we harness additional well-known linguistic heuristics. Named entities occur often in plural and genitive forms, i.e., AGDISTIS tries to identify and stem those words. For example, the genitive form of the named entity `Obama's` is transformed into `Obama`. Additionally, AGDISTIS reduces plural strings such as `Obamas` to the singular form `Obama`. Another heuristic is to remove common affixes. For example, we remove affixes which stands for the form of enterprises, such as *corp* and *ltd*, e.g., `Hanover Insurance Corp.` is shrunk to `Hanover Insurance` in order to find candidates for this string in the KB. AGDISTIS also eliminates candidates with years and dates within the label so as to be time-independent and to prune the search space. One key advantage of Linked Data is the possibility to retrieve a class for each instance in a KB. By using entity types (obtained via the `rdf:type` property), a domain fitting of possible candidates is implemented, narrowing the search space. Since our goal is to disambiguate persons, organizations and places, AGDISTIS only allows candidates of the types mentioned in Table 1 when run on DBpedia. Obviously, these classes can be altered by the user as required to fit his purposes. As knowledge base we chose DBpedia as a timely and popular source

Table 1. DBpedia classes used for disambiguation classes

Class	`rdf:type`
Person	dbpedia-owl:Person, foaf:Person
Organization	dbpedia-owl:Organization, dbpedia-owl:WrittenWork (e.g., Journals)
Place	dbpedia-owl:Place, yago:YagoGeoEntity

of Linked Data covering many different domains. Furthermore, it is straightforward to use `dbpedia-owl:wikiPageRedirects` for identifying multiple labels for one instance. Of course AGDISTIS ignores disambiguation pages as they would not help accomplishing the disambiguation goal and finding $\mu^\star$. Finally, our system compares the heuristically obtained label with the label extracted from the knowledge base by using *trigram similarity*. In order to find an optimal disambiguation accuracy we iterate the similarity threshold σ (cf. Section 5.1).

4.3 Disambiguation algorithm

Given a set of candidate nodes, we need to construct a semantic context graph using the underlying knowledge base K. In our case, we build the graph $G = (V, E)$ using the background KB. Let E be edges of existing ontology properties and V entity nodes from K. We employ a breadth-first search (bfs) for exploring the semantic space with depth d and nodes C as initial nodes set. Empirically, we see no effect on the accuracy when using spread activation instead [11] (despite the obvious extra computational costs).

After constructing the semantic context graph we need to identify the correct candidate node for a given named entity. Using the graph-based HITS algorithm [12] we calculate the most authoritative candidates by assigning the following values for each node iteratively:

$$x_a \longleftarrow \sum_{(y,x)\in E} y_h \tag{1}$$

$$y_h \longleftarrow \sum_{(y,x)\in E} x_a \tag{2}$$

We choose the number of iterations k according to Kleinberg [12], i.e., 20 iterations, which suffice to achieve converged authoritative values x_a and hub values y_h. Afterwards we sort the nodes according to their authoritative values in descending order. The first candidate for a certain named entity is assumed to be the correct disambiguation. AGDISTIS' whole procedure is presented in Algorithm 3. For our example, the graph depicted in Figure 2 shows an excerpt of the input graph for the HITS disambiguation algorithm when relying on DBpedia as knowedge base. The results can be seen in Table 2. Obviously a disambiguation towards the correct named entity URIs is possible.

Algorithm 3: Disambiguation Algorithm based on HITS and Linked Data.

Data: $N = \{N_1, N_2 \dots N_n\}$ named entities, σ trigram similarity threshold
Result: $C = \{C_1, C_2 \dots C_n\}$ identified candidates for named entities
begin
  $V \longleftarrow \emptyset, E \longleftarrow \emptyset, d \longleftarrow 2, k \longleftarrow 20, G \longleftarrow (V, E)$;
  $V \longleftarrow$ **insertCandidates(N,** σ**);**
  breadthFirstSearch(G, d);
  HITS(G, k);
  sortAccordingToAuthorityValue(V);
  for $N_i \in N$ **do**
    for $v \in V$ **do**
      if v **is a candidate for** N_i **then**
        store(N_i**,**v**);**
        break;

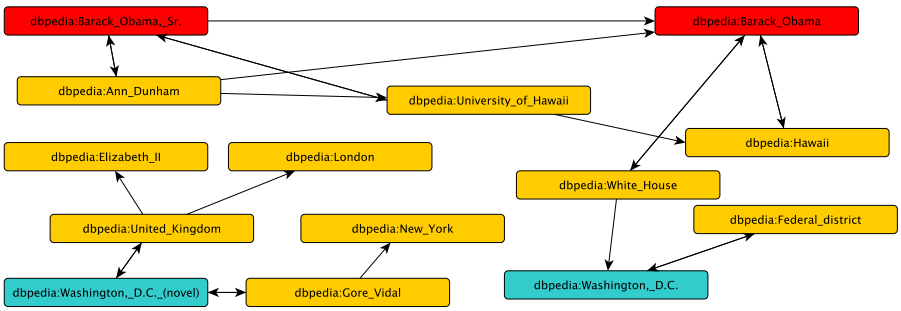

Fig. 2. One possible graph for the example sentence, with candidate nodes in red and blue.

Table 2. Authority weights for example graph.

Node	Authority Weight
dbpedia:Barack_Obama,_Sr.	0.089
dbpedia:Barack_Obama	0.273
dbpedia:Washington,_D.C._(novel)	0.000
dbpedia:Washington,_D.C.	0.093

5 Evaluation

5.1 Experimental Setup

The goal of our evaluation was to measure the accuracy achieved by our approach on different datasets. Moreover, we wanted to know how AGDISTIS performs in comparison to the state-of-the-art in NED. To achieve this goal, we ran AGDISTIS with the following parameter settings: the threshold σ for the trigram similarity was varied between 0 and 1 in steps of 0.01. Additionally, we tried $d = 1, 2, 3$ to account for the size of the semantic context graph and its influence on the accuracy. We also ran the experiments without using the graph, i.e., only applying all heuristics and trigram similarity. We did not considered abbreviations and thus ignored labels shorter than three characters. Moreover a closed-world was assumed, i.e., entities not in the KB were not considered in our evaluation. We carried out all our experiments on three corpora: (1) a subset of the well-known Reuters-21578 dataset, (2) RSS feeds extracted from 1500 sources and (3) a German news corpus extracted from `news.de`. For each corpus, we generated a spell-corrected version of annotations. While annotating we also left out ticker symbols of companies or job descriptions, such as *GOOG* for Google Inc. or minister of defense, because those are always preceded by the full company name respectively a persons name. Since AGDISTIS is based upon a closed-world assumption, we generated a default URI for instances which could not be identified within a 5 minutes web search while annotating. The number of documents and entities annotated is presented in Table 3. The test corpora can be downloaded from `https://github.com/AKSW/AGDISTIS`. We only considered `rdfs:label` as labeling property.

Table 3. Test corpora specifications

Corpus	Language	# Documents	# Entities	Annotation source
Reuters-21578	English	145	769	voter agreement
RSS 500	English	500	1000	domain expert
`news.de`	German	53	627	domain expert

Reuters-21578 Dataset. To create annotations of disambiguated named entities, we implemented a supporting judgment tool (see Figure 3).

The input for the tool was a subset of 145 Reuters-21578 news articles sampled randomly. First, FOX [15] was used for recognizing named entities. Second, we highlighted the found entities and added initial candidates via simple string matching algorithms. Our tool allows for disambiguating arbitrary texts by choosing candidates from the middle or selecting non-detected named entities. Two domain expert judges, each voting over 1,000 times, evaluated the texts. The voter agreement was 74%. In cases where the judges did not agree initially

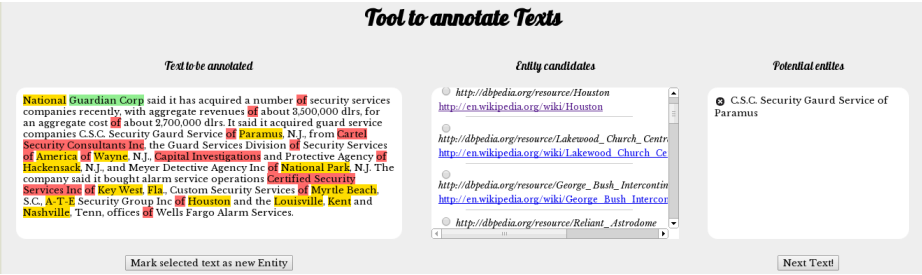

Fig. 3. GUI of our annotation tool.

a repeated check lead to an achieved agreement. This low initial agreement rate reveals the difficulty of the disambiguation task.

news.de Dataset. This dataset was collected from 2009 to 2011 from `news.de` ensuring that each message contains the German word *Golf*. This word is a homonym that can semantically mean a geographical gulf, a car model or the sport discipline. We used 53 texts comprising over 600 named entities that were annotated manually by a domain expert.

RSS 500 Dataset. To generate this corpus, we used a list of 1,457 RSS feeds as compiled in [8]. The list includes all major worldwide newspapers and a wide range of topics, e.g., *World*, *U.S.*, *Business*, *Science* etc. This list was crawled for 76 hours, which resulted in a corpus of about 11.7 million sentences. We created a subset of this corpus by randomly selecting 1% of the contained sentences. Finally three domain experts annotated 500 sentences manually. These sentences were a subset of those which contained a natural language representation of a formal relation, like "..., who was born in..." for `dpo:birthPlace` (see [7]), that occurred more then 5 times in the 1% corpus.

5.2 Results

Influence of Depth and Similarity Overall, AGDISTIS performs best on the `news.de` corpus (see Figure 4), where it achieves a maximal accuracy 0.72 for $\sigma = 0.81$ and $d = 2$ (see Table 4). Our approach achieves comparable results on the Reuters-21578 corpus (see Figure 5), where it reaches its peak 0.69 for $\sigma = 0.81$ and $d = 3$. In combination with the results on RSS 500 (see Figure 6), our results suggest that a $d = 2$ and $\sigma = 0.81$ are a good setting for AGDISITIS and suffice to perform well. In the only case where $d = 3$ leads to better results (Reuters-21578 corpus), the setting $d = 2$ is only outperformed by 0.01% accuracy. The good performance of AGDISTIS on the `news.de` dataset reveals the advantages of agnostic approach, as depicted in Figure 4. Note that the German `news.de` dataset is particularly difficult since each text contains the

Table 4. Evaluation of Datasets with LODNED against AIDA. σ stands for the trigram similarity.

Corpus	**AGDISTIS**			**AIDA**
	Accuracy	σ	d	Accuracy
Reuters-21578	0.69	0.83	3	0.49
RSS 500	0.60	0.81	2	0.57
`news.de`	0.72	0.81	2	—

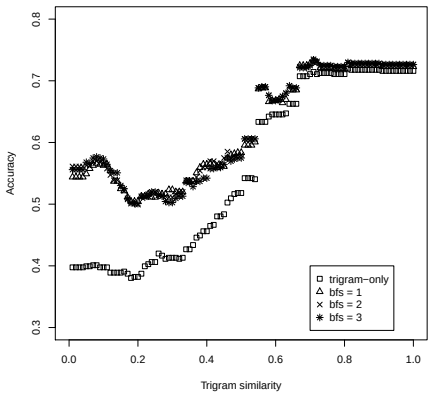

Fig. 4. Accuracy of AGDISTIS on the **news.de** corpus.

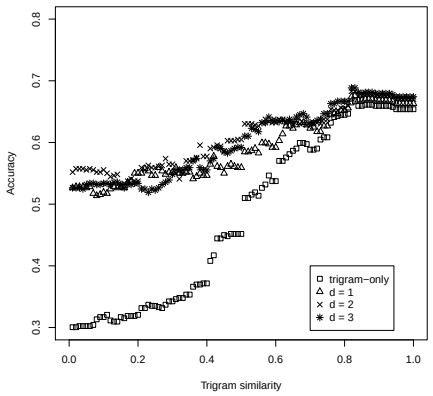

Fig. 5. Accuracy of AGDISTIS on the Reuters-21578 corpus.

German homonym *Golf* as explained before. Varying the search depth d does not significantly improve accuracy because within the underlying documents there are many similar named entities forming a shallow semantic background. Concerning the expansion policy there are two cases: either the first entity and its expansions are disambiguated correctly or the wrong disambiguation of the first entity leads to an avalanche of false evaluations. Still, reaching an accuracy of ≈ 0.72 reveals the capability of performing well independently of the underlying language. We observed a significant enhancement of AGDISTIS by using surface forms, as explained in Section 4.2. Employing additional labels, such as surface forms, increased the accuracy of AGDISTIS by up to 4% as shown in Figure 7.

Comparison with AIDA. We compared our approach to *AIDA* [9], a state-of-the-art NED algorithm using the *Cocktailparty configuration*, i.e., recommended configuration options of AIDA.[10] The results of this evaluation on AIDA can be seen in Table 4. Overall, AIDA performs well on arbitrary entities. Yet, it

[10] `https://github.com/yago-naga/aida`

is clearly outperformed by our approach on specific persons and organizations. In comparison to AIDA , AGDISTIS performs best on the Reuters-21578 where it surpasses AIDA by $\approx 20\%$ accuracy. Note that AGDISTIS also outperforms AIDI for our overall default setting of $\sigma = 0.81$. Furthermore, AGDISTIS outperforms AIDA with the RSS 500 corpus (see Fig. 6) by 3% accuracy. This corpus differs considerably from the Reuters-21578 corpus due to the small disambiguation contexts and graphs evolving from two named entities per text only. AIDA could not be run on the `news.de` corpus as it can only deal with English. Here, the language-independence of AGDISTIS pays off.

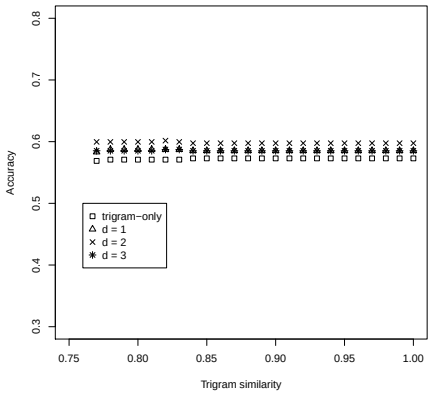

Fig. 6. Accuracy of AGDISTIS on the RSS 500 corpus.

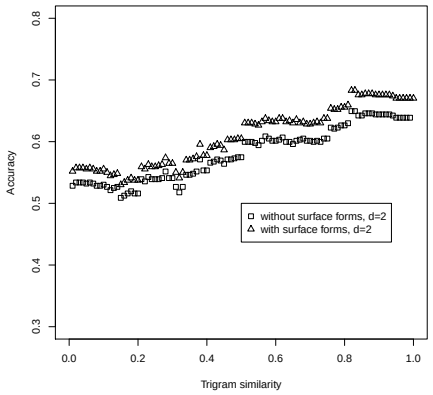

Fig. 7. Influence of surface forms on the accuracy.

Influence of Structure of Underlying Knowledge Base One of the main ideas behind AGDISTIS was to create an approach that performs well on the wide number of knowledge bases available on the Web of Data. Thus, we were interested in quantifying the influence of the topology of underlying knowledge base on AGDISTIS. To achieve this goal, we analyzed the influence of particular properties on AGDISTIS' accuracy performance. We computed the permutation $\Pi^\star$ of the most important properties $p_1, p_2, \ldots, p_n = P, p_i \in K$ w.r.t. accuracy by subsequent removal from K. Formally:

$$\Pi^\star = \underset{\Pi \in PERM(n)}{\arg\min} \quad \text{accuracy}(P, \Pi) \tag{3}$$

The results of this experiment on DBpedia and the Reuters-21578 corpus are shown in Table 5. Our results suggest that domain-specific properties have the most influence on our approach. For example, the first three properties `owner`,

Table 5. Effect of cumulatively removing properties, trigram similarity threshold $\sigma = 0.835$, Reuters-21578, $d = 2$, Prefix: dbpedia-owl.

Property	Accuracy
owner	0.675
isPartOf	0.672
countySeat	0.670
city	0.666
owningCompany, party, hometown, religion, memberOfParliament, occupation, influencedBy, locationCity, type, deputy, aircraftHelicopter, spouse, usingCountry, architect, knownFor, place, residence, governor, industry, timeZone, influenced, league, division, deathPlace, team, populationPlace, keyPerson, bandMember, operatedBy, officialLanguage, broadcastNetwork, march, education, season, regionalLanguage, battle, aircraftRecon, stylisticOrigin, militaryUnit, predecessor, wikiPageDisambiguates, leaderFunction, youthWing, derivative, fourthCommander, nationality, leaderParty, languageRegulator, leftTributary, parentCompany, language, country, militaryBranch, mayor, athletics, languageFamily, formerBandMember, largestCity, programmeFormat, state, location, territory, musicSubgenre, affiliation, aircraftTransport, associatedBand, garrison, primeMinister, part, otherParty, restingPlace, locationCountry, currency, aircraftElectronic, foundationPlace, related, neighboringMunicipality, award, relation, chairman, foundedBy, distributingLabel, isPartOfMilitaryConflict, colour, secondCommander, commander, ideology, lieutenant, formerTeam, broadcastArea, product, commandStructure, child, ground, routeStart, headquarter, spokenIn, profession, musicFusionGenre, recordLabel, anthem, successor, notableCommander, regionServed, governingBody, operator, service, hubAirport, militaryRank, aircraftTrainer, president	0.664
parent, jurisdiction, vicePresident, district, genre, pictureFormat, capital, birthPlace, region, governmentType, leaderName, leader, ethnicGroup, targetAirport, campus, monarch, thirdCommander, subsidiary, twinCity, almaMater, instrument, internationalAffiliation	0.661
managerClub, manager	0.663
sisterStation	0.664

`isPartOf`, `countySeat` are often linked to companies. Given that the most entities in the Reuters-21578 corpus stem from a business background, the removal of these properties has the highest influence on AGDISTIS' accuracy. The second and third group of generic properties does not influence our results significantly. Most interestingly, the removal of certain properties has the potential to improve our results. This is most probably due to these properties adding noise to authority scores generated by HITS and thus leading to a wrong ordering of the resources. Overall, our results suggest that Linked Data sources mostly boost disambiguation results through the domain-specific predicates. Thus, devising means to recognizing the domain at hand, mapping the properties from the underlying knowledge base that are of relevance for this domain can boost the accuracy of NED. Moreover, approaches for discarding noisy properties can also improve the accuracy of NED approaches. This observation also has an impact on the design of Linked Data sources for semantic search as it implies that by following `owl:sameAs` links and using domain-specific relations between resources across the whole Linked Data Web, we should be able to further improve the results achieved by AGDISTIS.

6 Conclusion

We presented AGDISTIS a novel, LOD-background-knowledge-based named entity disambiguation approach that outperforms the state-of-the-art algorithm AIDA. Since AGDISTIS is agnostic of the underlying knowledge base, it can profit from growing KBs as well as multilingual Linked Data. By combining the Web-scale HITS algorithm and breadth-first search with linguistic heuristics, we were able to precisely disambiguate a wide range of named entities. Furthermore, we measured the effect of evolving the structure of the underlying knowledge base. We observed the significance of properties on the accuracy of our system. Our results suggest that only a few RDF properties contribute significantly to enhancing the performance of AGDISTIS. We see this work as the first step in a larger research agenda. Based on AGDISTIS, we aim to establish a new paradigm of realizing NLP services, which employ community-generated, multilingual and evolving LOD background knowledge. Other than most work, which mainly uses statistics and heuristics, we aim to truly exploit the graph structure and semantics of the background knowledge. In the future we intend to look for larger and even more insightful disambiguation datasets to refine and test AGDISTIS. Moreover, a deeper evaluation of ontology structures towards disambiguation accuracies is needed. Answering those research questions will expose possible performance-enhancing extensions.

References

1. Benjamin Adrian, Jörn Hees, Ivan Herman, Michael Sintek, and Andreas Dengel. Epiphany: Adaptable rdfa generation linking the web of documents to the web of data. In Philipp Cimiano and H.Sofia Pinto, editors, *Knowledge Engineering and Management by the Masses*, volume 6317 of *Lecture Notes in Computer Science*, pages 178–192. Springer Berlin Heidelberg, 2010.
2. Sören Auer, Chris Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web Conference (ISWC)*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer, 2008.
3. Paolo Bouquet, Heiko Stoermer, Giovanni Tummarello, and Harry Halpin, editors. *Proceedings of the WWW2007 Workshop I^3: Identity, Identifiers, Identification, Entity-Centric Approaches to Information and Knowledge Management on the Web, Banff, Canada, May 8, 2007*, volume 249 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.
4. Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, pages 708–716, 2007.
5. Basil Ell, Denny Vrandečic, and Elena Simperl. Labels in the web of data. In *Proceedings of the 10th international conference on The semantic web - Volume Part I*, ISWC'11, pages 162–176, Berlin, Heidelberg, 2011. Springer-Verlag.
6. Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

7. Daniel Gerber and Axel-Cyrille Ngonga Ngomo. Extracting Multilingual Natural-Language Patterns for RDF Predicates. In *EKAW*, Lecture Notes in Computer Science. Springer, 2012.
8. Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, 2012.
9. Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust Disambiguation of Named Entities in Text. In *Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, Edinburgh, Scotland*, pages 782–792, 2011.
10. Joachim Kleb and Andreas Abecker. Entity reference resolution via spreading activation on rdf-graphs. In *ESWC (1)*, pages 152–166, 2010.
11. Joachim Kleb and Andreas Abecker. Disambiguating entity references within an ontological model. In *WIMS*, page 22, 2011.
12. Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, September 1999.
13. Pablo N. Mendes, Max Jakob, Andres Garcia-Silva, and Christian Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*, 2011.
14. David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30:3–26, 2007.
15. Axel-Cyrille Ngonga Ngomo, Norman Heino, Klaus Lyko, René Speck, and Martin Kaltenböck. Scms–semantifying content management systems. In *The Semantic Web–ISWC 2011*, pages 189–204. Springer, 2011.
16. Bruno Pouliquen, Ralf Steinberger, Camelia Ignat, Emilia Käsper, and Irina Temnikova. Multilingual and cross-lingual news topic tracking. In *Proceedings of the 20th international conference on Computational Linguistics*, page 959. Association for Computational Linguistics, 2004.
17. L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *CoNLL*, 6 2009.
18. Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1375–1384, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.
19. Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. Large-scale cross-document coreference using distributed inference and hierarchical models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 793–803. Association for Computational Linguistics, 2011.
20. Gang Wang, Yong Yu, and Haiping Zhu. Pore: Positive-only relation extraction from wikipedia text. In *The Semantic Web*, pages 580–594. Springer, 2007.