AGDISTIS - Graph-Based Disambiguation of Named Entities using Linked Data

Ricardo Usbeck[1,2], Axel-Cyrille Ngonga Ngomo[1], Michael Röder[1,2], Daniel Gerber[1], Sandro Athaide Coelho[3], Sören Auer[4], and Andreas Both[2]

[1] University of Leipzig, Germany , [2] R & D, Unister GmbH, Germany, [3] Federal University of Juiz de Fora, Brazil, [4] University of Bonn & Fraunhofer IAIS, Germany
email: {usbeck|ngonga}@informatik.uni-leipzig.de

Abstract. Over the last decades, several billion Web pages have been made available on the Web. The ongoing transition from the current Web of unstructured data to the Web of Data yet requires scalable and accurate approaches for the extraction of structured data in RDF (Resource Description Framework) from these websites. One of the key steps towards extracting RDF from text is the disambiguation of named entities. While several approaches aim to tackle this problem, they still achieve poor accuracy. We address this drawback by presenting AGDISTIS, a novel knowledge-base-agnostic approach for named entity disambiguation. Our approach combines the Hypertext-Induced Topic Search (HITS) algorithm with label expansion strategies and string similarity measures. Based on this combination, AGDISTIS can efficiently detect the correct URIs for a given set of named entities within an input text. We evaluate our approach on eight different datasets against state-of-the-art named entity disambiguation frameworks. Our results indicate that we outperform the state-of-the-art approach by up to 29% F-measure.

1 Introduction

The vision behind the Web of Data is to provide a new machine-readable layer to the Web where the content of Web pages is annotated with structured data (e.g., RDFa [1]). However, the Web in its current form is made up of at least 15 billion Web pages.[1] Most of these websites are unstructured in nature. Realizing the vision of a usable and up-to-date Web of Data thus requires scalable and accurate natural-language-processing approaches that allow extracting RDF from such unstructured data. Three tasks play a central role when extracting RDF from unstructured data: named entity recognition (NER), named entity disambiguation (NED), also known as entity linking [16], and relation extraction (RE). For the first sentence of Example 1, an accurate named entity recognition approach would return the strings `Barack Obama` and `Washington, D.C.`. A high-quality DBpedia-based named entity disambiguation (NED) approach would use these already recognized named entities and map the strings

[1] Data gathered from `http://www.worldwidewebsize.com/` on January 4th, 2014.

`Barack Obama` resp. `Washington, D.C.` to the resources `dbr:Barack_Obama` and `dbr:Washington,_D.C.`[2] [14].

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

While NER has been explored extensively over the last decades [7], the disambiguation of named entities, i.e., the assignment of a resource's URI from an existing knowledge base to a string that was detected to label an entity remains a difficult task.

Current NED approaches suffer from two major drawbacks: First, they poorly perform on Web documents [20]. This is due to Web documents containing resources from different domains within a narrow context. An accurate processing of Web data has yet been shown to be paramount for the implementation of the Web of Data [8]. Well-know approaches such as *Spotlight* [15] and *TagMe 2* [6] have been designed to work on a particular knowledge base. However, Web data contains resources from many different domains. Hence, we argue that NED approaches have to be designed in such a way that they are agnostic of the underlying knowledge base. Second, most state-of-the-art approaches rely on exhaustive data mining methods [4,21] or algorithms with non-polynomial time complexity [11]. However, given the large number of entities that must be disambiguated when processing Web documents, scalable NED approaches are of central importance to realize the Semantic Web vision.

In this paper, we address these drawbacks by presenting AGDISTIS, a novel NED approach and framework. AGDISTIS achieves *higher F-measures* than the state of the art while remaining *polynomial in its time complexity*. AGDISTIS achieves these results by combining the HITS algorithm [12] with label expansion and string similarity measures. Overall, our contributions can be summed up as follows: (1) We present AGDISTIS, an accurate and scalable framework for disambiguating named entities that is agnostic to the underlying knowledge base (KB) and show that we are able to outperform the state of the art by up to 29% F-measure on these datasets. (2) We show that our approach has a quadratic time complexity. Thus, it scales well enough to be used even on large knowledge bases. (3) We evaluate AGDISTIS on eight *well-known and diverse open-source datasets.*[3]

The rest of this paper is organized as follows: We first give a brief overview of related work in Section 2. Then, we introduce the AGDISTIS approach in Section 3. After presenting the datasets, we evaluate our approach against the state of the art frameworks AIDA and TagMe 2 and the well-known DBpedia Spotlight. Furthermore, we measure the influence of using surface forms, i.e., synonymous label for a specific resource, in Section 4. We conclude in Section 5 by highlighting research questions that emerged from this work. A demo of our

[2] `dbr:` stands for `http://dbpedia.org/resource/`

[3] Further data, detailed experimental results and source code for this paper are publicly available on our project homepage `http://aksw.org/Projects/AGDISTIS`.

approach (integrated into the Named Entity Recognition framework FOX [25]) can be found at `http://fox.aksw.org`.

2 Related Work

AGDISTIS is related to the research area of Information Extraction [19] in general and to NED in particular. Several approaches have been developed to tackle NED. Cucerzan presents an approach based on extracted Wikipedia data towards disambiguation of named entities [4]. The author aims to maximize the agreement between contextual information of Wikipedia pages and the input text by using a local approach. *Epiphany* [2] identifies, disambiguates and annotates entities in a given HTML page with RDFa. Ratinov et al. [21] described an approach for disambiguating entities from Wikipedia KB. The authors argue that using Wikipedia or other ontologies can lead to better global approaches than using traditional local algorithms which disambiguate each mention separately using, e.g., text similarity. 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 use Wikipedia and other Linked Data KBs. LINDEN [23] is an entity linking framework that aims at linking identified named entities to a knowledge base. To achieve this goal, LINDEN collects a dictionary of the surface forms of entities from different Wikipedia sources, storing their count information.

Wikipedia Miner [17] is the oldest approach in the field of *wikification*. Based on different machine learning algorithms, the systems disambiguates w.r.t. prior probabilities, relatedness of concepts in a certain window and context quality. The authors evaluated their approach based on a Wikipedia as well as an AQUAINT subset. Unfortunately, the authors do not use the opportunities provided by Linked Data like DBpedia.

Using this data the approach constructs candidate lists and assigns link probabilities and global coherence for each resource candidate. The AIDA approach [9] for NED tasks is based on the YAGO2[4] knowledge base and relies on 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.

Another approach is DBpedia Spotlight [15], a framework for annotating and disambiguating Linked Data Resources in arbitrary texts. In contrast to other tools, Spotlight is able to disambiguate against all classes of the DBpedia ontology. Furthermore, it is well-known in the Linked Data community and used in various projects showing its wide-spread adoption.[5] Based on a vector-space model and cosine similarity DBpedia Spotlight is publicly available via a web service[6].

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

[5] `https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki/Known-uses`

[6] `https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki/Web-service`

In 2012, Ferragina et al. published a revised version of their disambiguation system called TagMe 2. The authors claim that it is tuned towards smaller texts, i.e., comprising around 30 terms. TagMe 2 is based on an anchor catolog (`<a>` tags on Wikipedia pages with a certain frequency), a page catalogue (comprising all original Wikipedia pages, i.e., no disambiguations, lists or redirects) and an in-link graph (all links to a certain page within Wikipedia). First, TagMe 2 identifies named entities by matching terms with the anchor catalog and second disambiguates the match using the in-link graph and the page catalog via a collective agreement of identified anchors. Last, the approach discards identified named entities considered as non-coherent to the rest of the named entities in the input text.

In 2014, Babelfy [18] has been presented to the community. Based on random walks and densest subgraph algorithms Babelfy tackles NED and is evaluated with six datasets, one of them the later here used AIDA dataset. In constrast to AGDISTIS, Babelfy differentiates between word sense disambiguation, i.e., resolution of polysemous lexicographic entities like *play*, and entity linking, i.e., matching strings or substrings to knowledge base resources. Due to its recent publication Babelfy is not evaluated in this paper.

Recently, Cornolti et al. [3] presented a benchmark for NED approaches. The authors compared six existing approaches, also using DBpedia Spotlight, AIDA and TagMe 2, against five well-known datasets. Furthermore, the authors defined different classes of named entity annotation task, e.g. *'D2W'*, that is the disambiguation to Wikipedia task which is the formal task AGDISITS tries to solve. We consider TagMe 2 as state of the art w.r.t. this benchmark although only one dataset has been considered for this specific task. We analyze the performance of DBpedia Spotlight, AIDA, TagMe 2 and our approach AGDISTIS on four of the corpora from this benchmark in Section 4.

3 The AGDISTIS Approach

3.1 Named Entity Disambiguation

The goal of AGDISTIS is to detect correct resources from a KB K for a vector N of n a-priori determined named entities $N_1, \ldots, N_n$ extracted from a certain input text T. In general, several resources from a given knowledge base K can be considered as candidate resources for a given entity N_i. For the sake of simplicity and without loss of generality, we will assume that each of the entities can be mapped to m distinct candidate resources. Let C be the matrix which contains all candidate-entity mappings for a given set of entities. The entry C_{ij} stands for the j^{th} candidate resource for the i^{th} named entity. 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 vector of resources of length $|N|$ that is such that the i^{th} entry of the vector maps with N_i.

Let ψ be a function which computes the similarity between an assignment $\mu(C, N)$ and the vector of named entities N. The *coherence* function ϕ calculates the similarity of the knowledge base K and an assignment μ, cf. Ratinov et al. [21], to ensure the topical consistency of μ. The coherence function ϕ is implemented by the HITS algorithm, which calculates the most pertinent entities while the similarity function ψ is, e.g., string similarity. Given this formal model, the goal is to find the assignment $\mu^\star$ with

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

The formulation of the problem given above has been proven to be NP-hard, cf. Cucerzan et al. [4]. Thus, for the sake of scalability, AGDISTIS computes an approximation μ^+ by using HITS, a fast graph algorithm which runs with an upper bound of $\Theta(k \cdot |V|^2)$ with k the number of iterations and $|V|$ the number of nodes in the graph. Furthermore, using HITS leverages 1) scalability, 2) well-researched behaviour and 3) the ability to explicate semantic authority.

3.2 Architecture

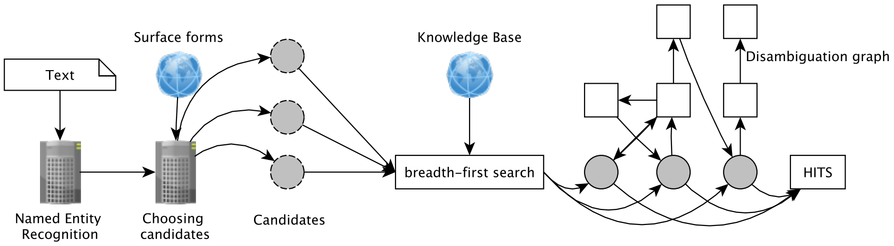

Fig. 1: Overview of AGDISTIS.

Our approach to NED thus consists of three main phases as depicted in Figure 1. Given an input text T and a named entity recognition function (e.g., FOX [25]), 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 [15] for resources from the underlying KB. The set of candidates generated by the first step is used to generate a disambiguation graph. Here, we rely on a graph search algorithm which retrieves context information from the underlying KB. Finally, we employ the HITS algorithm to the context graph to find authoritative candidates for the discovered named entities. We assume that the resources with the highest authority values represent the correct candidates. All algorithms in AGDISTIS have a polynomial time complexity, leading to AGDISTIS also being polynomial in time complexity. Choosing candidates relates to the notion of ϕ while calculating the authority values confers to ψ. In the following, we present each of the steps of AGDISTIS in more detail.

3.3 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 KB. We begin by creating an index comprising all labels of each resource. Our approach can be configured to use any set of properties as labeling properties (e.g., those in Ell et al. [5]). For our experiments, we only considered `rdfs:label` as labeling property. In addition, our approach can make use of known *surface forms* for each of the resources in case such knowledge is available [15]. These are simply strings that are used on the Web to refer to given resources. Surface forms are simply added to the set of available labels for each resource, cf. Section 4.1. In this paper, we do not consider abbreviations although these could be easily regarded by adding further labels into the KB (e.g., via WordNet[7]).

Next to searching the index we apply a *string normalization* approach and an *expansion policy* to the input text: The string normalization is 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. For example, the genitive `New York's` is transformed into `New York`, the postfix of `Microsoft Ltd.` is reduced to `Microsoft` and the time information of `London 2013` is ignored. Our *expansion policy* is a time-efficient approach to coreference resolution, which plays a central role when dealing with text from the Web, cf. Singh et al. [24]. In web and news documents, named entities are commonly mentioned in their full length the first time they appear, while the subsequent mentions only consist of a substring of the original mention due to the brevity of most news data. For example, a text mentioning Barack Obama's arrival in Washington D.C. will commonly contain `Barack Obama` in the first mention of the entity and use strings such as `Obama` or `Barack` later in the same text (see Example 1). We implement this insight by mapping each named entity label (e.g., `Obama`) which is a substring of another named entity label that was recognized previously (e.g., `Barack Obama`) to the same resource , i.e., `dbr:Barack_Obama`. If there are several possible expansions, we choose the shortest as a fast coreference resolution heuristic for web documents. Without the expansion policy AGDISTIS suffers from a loss of accuracy of $\approx 4\%$.

Additionally, AGDISTIS can be configured to fit named entities to certain domains to narrow 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 and YAGO2. Adding general types will increase the number of candidates and thus decrease the performance. Obviously, these classes can be altered by the user as required to fit his purposes.

The resulting candidate detection approach is explicated in Algorithm 1. In its final step, our system compares the heuristically obtained label with the label extracted from the KB by using *trigram similarity* which is an n-gram similarity with $n = 3$.

[7] `http://wordnet.princeton.edu/`

Table 1: DBpedia and YAGO2 classes used for disambiguation classes. Prefix `dbo` stands for `http://dbpedia.org/ontology/`, `foaf` for `http://xmlns.com/foaf/0.1/` and `yago` for `http://yago-knowledge.org/resource/`.

	Class	`rdf:type`
DBpedia	Person	dbo:Person, foaf:Person
DBpedia	Organization	dbo:Organization, dbo:WrittenWork (e.g., Journals)
DBpedia	Place	dbo:Place, yago:YagoGeoEntity
YAGO2	Person	yago:yagoLegalActor
YAGO2	Organization	yago:yagoLegalActor, yago:wordnet_exchange_111409538 (e.g., NASDAQ)
YAGO2	Place	yago:YagoGeoEntity

Algorithm 1: Searching candidates for a label.

```
Data: label of a certain named entity $N_i$, $\sigma$ trigram similarity threshold
Result: $C$ candidates found
$C \longleftarrow \emptyset$;
label ⟵ normalize(label);
label ⟵ expand(label);
$\bar{C} \longleftarrow$ searchIndex(label);
for c $\in \bar{C}$ do
    if $\neg$c .matches([0-9]$^+$) then
        if trigramSimilarity(c, label)$\geq \sigma$ then
            if fitDomain(c) then
                $C \longleftarrow C \cup$ c;
```

3.4 Computation of Optimal Assignment

Given a set of candidate nodes, we begin the computation of the optimal assignment by constructing a disambiguation graph G_d with search depth d. To this end, we regard the input knowledge base as a directed graph $G_K = (V, E)$ where the vertices V are resources of K, the edges E are properties of K and $x, y \in V, (x, y) \in E \Leftrightarrow \exists p : (x, p, y)$ is an RDF triple in K. Given the set of candidates C, we begin by building an initial graph $G_0 = (V_0, E_0)$ where V_0 is the set of all resources in C and $E_0 = \emptyset$. Starting with G_0 we extend the graph in a breadth-first search manner. Therefore, we define the extension of a graph $G_i = (V_i, E_i)$ to a graph $\rho(G_i) = G_{i+1} = (V_{i+1}, E_{i+1})$ with $i = 0, \ldots, d$ as follows:

$$V_{i+1} = V_i \cup \{y : \exists x \in V_i \wedge (x, y) \in E\} \tag{1}$$

$$E_{i+1} = \{(x, y) \in E : x, y \in V_{i+1}\} \tag{2}$$

We iterate the ρ operator d times on the input graph G_0 to compute the initial disambiguation graph G_d.

After constructing the disambiguation graph G_d we need to identify the correct candidate node for a given named entity. Using the graph-based HITS algorithm we calculate authoritative values x_a, y_a and hub values x_h, y_h for all $x, y \in V_d$. We initialize the authoritative and hub values (3) and afterwards iterate the equations (4) k times as follows:

$$\forall x \in V_d, x_a = x_h = \frac{1}{|V_d|} \text{ (3) and } x_a \longleftarrow \sum_{(y,x)\in E_d} y_h, \quad y_h \longleftarrow \sum_{(y,x)\in E_d} x_a \text{(4)}$$

We choose k according to Kleinberg [12],i.e., 20 iterations, which suffice to achieve convergence in general. Afterwards we identify the most authoritative candidate C_{ij} among the set of candidates C_i as correct disambiguation for a given named entity N_i. When using DBpedia as KB and C_{ij} is a redirect AGDISTIS uses the target resource. AGDISTIS' whole procedure is presented in Algorithm 2. As can be seen, we calculate μ^+ solely by using polynomial time complex algorithms.

Algorithm 2: Disambiguation Algorithm based on HITS and Linked Data.

```
Data: $N = \{N_1, N_2 \dots N_n\}$ named entities, $\sigma$ trigram similarity threshold, $d$
      depth, $k$ number of iterations
Result: $C = \{C_1, C_2 \dots C_n\}$ identified candidates for named entities
$E \longleftarrow \emptyset$;
$V \longleftarrow$insertCandidates($N, \sigma$);
$G \longleftarrow (V, E)$;
$G \longleftarrow$breadthFirstSearch($G, d$);
HITS($G(V, E), k$);
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;
```

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 knowledge base. The results can be seen in Table 2.

4 Evaluation

4.1 Experimental Setup

The aim of our evaluation was two-fold. First, we wanted to determine the F-measure achieved by our approach on different datasets. Several definitions of

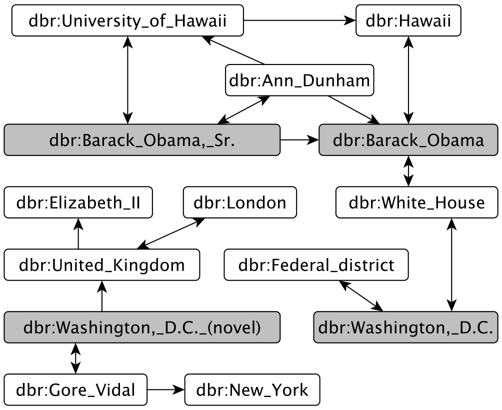

Fig. 2: One possible graph for the example sentence, with candidate nodes in grey.

Node	x_a
db:Barack_Obama	0.273
db:Barack_Obama,_Sr.	0.089
db:Washington,_D.C.	0.093
db:Washington,_D.C._(novel)	0.000

Table 2: Authority weights for example graph.

F-measure have been used in previous work on NED. Cornolti et al. [3] define the micro F-measure (F1) w.r.t. a strong annotation match (i.e., a binary relation) and the possibility of assigning null to an entity. This F-measure, which we use throughout our evaluation, aggregates all true/false positives/negatives over all documents. Thus, it accounts for larger contexts in documents with more annotations, cf. Cornolti et al. [3].

Second, we wanted to know how AGDISTIS performs in comparison to other state-of-the-art NED approaches. Thus, we compare AGDISTIS with TagMe 2, the best approach according to [3] as well as with AIDA and DBpedia Spotlight because they are well-known in the Linked Data community. AGDISTIS is designed to be agnostic of the underlying knowledge base. Thus, we use the German and English DBpedia KB as well as the English YAGO 2 KB.

Within our experiments, 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 evaluated our approach with $d = 1, 2, 3$ to measure the influence of the size of the disambiguation graph on AGDISTIS' F-measure. For our experiments, we fitted AGDISTIS to the domain of named entity recognition and only allow candidates of the types mentioned in Table 1. We report more details on the evaluation setup as well as complete results at the project homepage.

4.2 Datasets

Noisy and incorrect datasets can affect the performance of NED approaches which can be prevented by using well-known datasets. We carried out our evaluation on the following eight different, publicly available datasets, which consists of the three corpora from the benchmark dataset **N3** [22], the original AIDA evaluation corpus[8] and four of the five datasets from the Cornolti et al. [3] benchmark:

1. **Reuters-21578 Dataset.** The first of the N3 datasets comprises 145 news articles randomly sampled from the Reuters-21578 news articles dataset. Two domain experts determined the correct URI for each named entity using an online annotation tool reaching a initial voter agreement of 74%. In cases where the judges did not agree initially, they concerted each other and reached an agreement. This initial agreement rate hints towards the difficulty of the disambiguation task. The corpus does not annotate ticker symbols of companies (e.g., *GOOG* for Google Inc.), abbreviations and job descriptions because those are always preceded by the full company name respectively a person's name.
2. `news.de` **Dataset.** This real-world dataset is the second of the N3 datasets and was collected from 2009 to 2011 from the German web news portal `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. This dataset contains 53 texts comprising over 600 named entities that were annotated manually by a domain expert. Although some meanings of Golf are not within the class range of our evaluation, they are kept for evaluation purposes.
3. **RSS-500 Dataset.** This corpus has been published in Gerber et al. [8] and is the third of the of the N3 datasets. It consists of data scrapped from 1,457 RSS feeds. 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. A subset of this corpus has been created by randomly selecting 1% of the contained sentences. Finally, domain experts annotated 500 sentences manually. Further information about the corpora and the datasets themselves can be found on the project homepage.[9]
4. **AIDA-YAGO2 Dataset.** This is the original dataset that was used while evaluating AIDA [9], stemming from the CoNLL 2003 shared task [26] and comprising 1,393 news articles which were annotated manually.
5. **AIDA/CO-NLL-TestB** This dataset (like all the subsequent datasets) comes from the Cornolti et al. benchmarks and originates from the evaluation of AIDA [9]. As mentioned above, this dataset was derived from the CO-NLL 2003 shared task [26] and comprises 1,393 news articles which were annotated manually. Two students annotated each entity resolving conflicts by the authors of AIDA [9]. Cornolti et al.'s benchmark consists only of the second test part comprising 231 documents with 19.4 entities per document on average.
6. **AQUAINT** In this dataset, only the first mention of an entity is annotated. The corpus consists of 50 documents which are on average longer than the

[8] `https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/aida/downloads/`

[9] `http://aksw.org/Projects/N3NERNEDNIF.html`

Table 3: Test corpora specification including the number of documents (#Doc.) and the number of named entities (#Ent.) per dataset

Corpus	Language	#Doc.	#Ent.	Ent./Doc.	Annotation
AIDA/CO-NLL-TestB	English	231	4458	19.40	voter agreement
AQUAINT	English	50	727	14.50	voter agreement
IITB	English	103	11,245	109.01	domain expert
MSNBC	English	20	658	31.90	domain expert
Reuters-21578	English	145	769	5.30	voter agreement
RSS 500	English	500	1,000	2.00	domain expert
news.de	German	53	627	11.83	domain expert
AIDA-YAGO2	English	1,393	34,956	25.07	voter agreement

AIDA/CO-NLL-TestB documents. Each document contains 14.5 annotated elements on average The documents originate from different news services, e.g. Associated Press and have been annotated using voter agreement. The dataset was created by Milne et al. [17].

7. **IITB** The IITB corpus comprises 103 manually annotated documents. Each document contains 109.1 entities on average. This dataset displays the highest entity/document-density of all corpora. This corpus has been presented by Kulkarni et al. [13] in 2009.
8. **MSNBC** This corpus contains 20 news documents with 32.9 entities per document. This corpus was presented in 2007 by Cucerzan et al. [4].

We did not use the **Meij** dataset from Cornolti et al. since it comprises only tweets from twitter with 1.6 entities per document. The number of entities available in the datasets is shown in Table 3. All experiments were carried out on a MacBook Pro with a 2.7GHz Intel Core i7 processor and 4 GB 1333MHz DDR3 RAM using Mac OS 10.7.

4.3 Results

Table 4: Evaluation of AGDISTIS against AIDA and DBpedia Spotlight. Bold indicates best F-measure.

Corpus	**AGDISTIS**						**AIDA**	**Spotlight**
K	DBpedia			YAGO2			YAGO2	DBpedia
	F-measure	σ	d	F-measure	σ	d	F-measure	F-measure
Reuters-21578	**0.78**	0.87	2	0.60	0.29	3	0.62	0.56
RSS-500	**0.75**	0.76	2	0.53	0.82	2	0.60	0.56
news.de	**0.87**	0.71	2	—	—	—-	—-	0.84
AIDA-YAGO2	0.73	0.89	2	0.58	0.76	2	**0.83**	0.57

First, we evaluate AGDISTIS against AIDA and DBpedia Spotlight on three different knowledge bases using N3 corpora and the AIDA-YAGO2 corpus.

AGDISTIS performs best on the `news.de` corpus, achieving a maximal 0.87 F-measure for $\sigma = 0.71$ and $d = 2$ (see Table 4). Our approach also outperforms the state of the art on Reuters-21578 corpus (see Figure 3), where it reaches 0.78 F-measure for $\sigma = 0.87$ and $d = 2$. Considering the AIDA-YAGO2 dataset AGDISTIS achieves an F-measure of 0.73 for $\sigma = 0.89$ and $d = 2$. Our results suggest that $d = 2, \sigma = 0.82$ and using DBpedia as KB are a good setting for AGDISTIS and suffice to perform well. In the only case where $\sigma = 0.29$ leads to better results (Reuters-21578 corpus), the setting $0.7 < \sigma < 0.9$ is only outperformed by 0.03 F-measure using YAGO as KB for AGDISTIS.

Table 5: Performance of AGDISTIS, DBpedia Spotlight and TagMe 2 on four different datasets using micro F-measure (**F1**).

Dataset	Approach	**F1-measure**	**Precision**	**Recall**
AIDA/CO-NLL-TestB	TagMe 2	0.565	0.58	0.551
	DBpedia Spotlight	0.341	0.308	0.384
	AGDISTIS	**0.596**	**0.642**	**0.556**
AQUAINT	TagMe 2	0.457	0.412	**0.514**
	DBpedia Spotlight	0.26	0.178	0.48
	AGDISTIS	**0.547**	**0.777**	0.422
IITB	TagMe 2	0.408	0.416	0.4
	DBpedia Spotlight	**0.46**	0.434	**0.489**
	AGDISTIS	0.31	**0.646**	0.204
MSNBC	TagMe 2	0.466	0.431	0.508
	DBpedia Spotlight	0.331	0.317	0.347
	AGDISTIS	**0.761**	**0.796**	**0.729**

Second, we compared our approach with TagMe 2 and DBpedia using the datasets already implemented in the framework of Cornolti et al. AGDISTIS has been setup to use a breadth-first search depth $d = 2$ and a trigram similarity of $\sigma = 0.82$. All approaches used disambiguate w.r.t. the English DBpedia. AIDA was ommitted from this evaluation because it has been shown to be outperformed by TagMe 2 in [3] on the datasets we consider.

AGDISTIS achieves F-measures between 0.31 (IITB) and 0.76 (MSNBC) (see Table 5). We outperform the currently best disambiguation framework, TagMe 2, on three out of four datasets by up to 29.5% F-measure. Our poor performance on IITB is due to AGDISTIS not yet implementing a paragraph-wise disambiguation policy. By now, AGDISTIS performs disambiguation on full documents. The large number of resources in the IITB documents thus lead to our approach generating very large disambiguation graphs. The explosion of errors within these graphs results in an overall poor disambiguation. We will address this drawback in future work by fitting AGDISTIS with a preprocessor able to extract paragraphs from input texts. The local vector-space model used by Spotlight performs best in this setting.

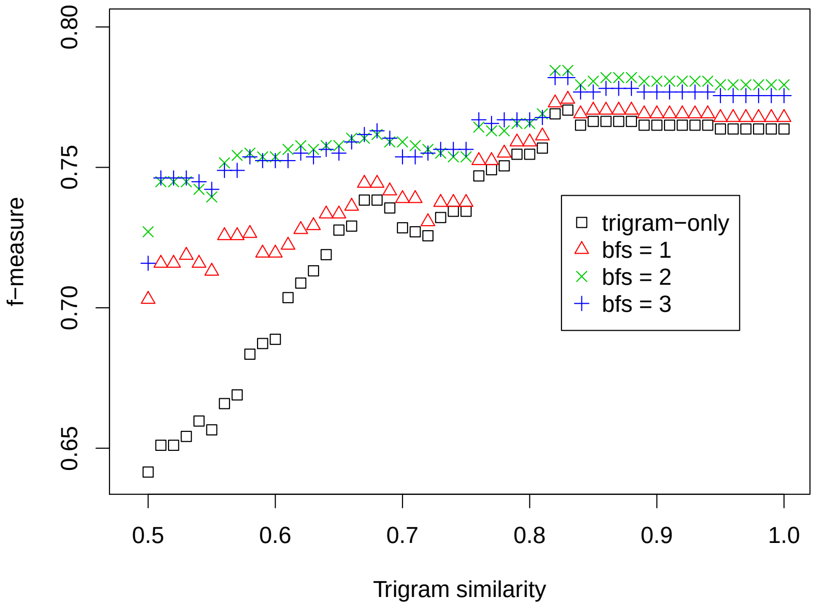

Fig. 3: F-measure on the **Reuters-21578** corpus using DBpedia as KB.

Delving deeper into AGDISTIS' results lead to the following insights: (1) Varying the search depth d does not significantly improve F-measure because within the underlying documents there are many similar named entities forming a shallow semantic background. However, using only string similarity measures ($d = 0$) results in lower F-measure (see Figure 3). (2) The expansion policy can have considerable knock-on effects: Either the first entity and its expansions are disambiguated correctly or the wrong disambiguation of the first entity leads to an avalanche of false results in a loss of $\approx 4\%$ accuracy. (3) We observed a significant enhancement of AGDISTIS when adding surface forms to the labels of resources as explained in Section 3.3. Employing additional labels (such as surface forms gathered from Wikipedia) increased the F-measure of AGDISTIS by up to 4%. (5) Using $n = 1, 2, 4$ as n-gram similarity has been proven to perform worse than using trigram similarity, i.e., $n = 3$. Our results suggest that $d = 2$ while using DBpedia as KB is a good setting for AGDISTIS and suffice to perform well. The iteration of σ between 0.7 and 0.9 can lead to an improvement of up to 6% F-measure while $\sigma < 0.7$ and $\sigma > 0.9$ leads to a loss of F-measure.

Overall, our results suggest that $\sigma = 0.82$ and $d = 2$ is generally usable across datasets and knowledge bases leading to high quality results.[10]

5 Conclusion

We presented AGDISTIS a novel named entity disambiguation that combines the scalable HITS algorithm and breadth-first search with linguistic heuristics. Our approach outperforms the state-of-the-art algorithms TagMe 2, AIDA and DBpedia Spotlight while remaining quadratic in its time complexity. Moreover, our evaluation suggests that while the approach performs well in and of itself, it can benefit from being presented with more linguistic information such as surface forms. We see this work as the first step in a larger research agenda. Based on AGDISTIS, we aim to develop a new paradigm for realizing NLP services which employ community-generated, multilingual and evolving Linked Open Data 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.

Since AGDISTIS is agnostic of the underlying knowledge base and language-independent, it can profit from growing KBs as well as multilingual Linked Data. In the future, we will thus extend AGDISTIS by using different underlying KBs and even more domain-specific datasets. An evaluation of Babelfy against our approach will be published on the project website. Moreover, we will implement a sliding-window-based extension of AGDISTIS to account for large amounts of entities per document.

Acknowledgments. Parts of this work were supported by the ESF and the Free State of Saxony and the FP7 project GeoKnow (GA No. 318159).

References

1. B. Adida, I. Herman, M. Sporny, and M. Birbeck. RDFa 1.1 Primer. Technical report, World Wide Web Consortium, http://www.w3.org/TR/2012/NOTE-rdfa-primer-20120607/, June 2012.
2. B. Adrian, J. Hees, I. Herman, M. Sintek, and A. Dengel. Epiphany: Adaptable rdfa generation linking the web of documents to the web of data. In *Knowledge Engineering and Management by the Masses*, pages 178–192. Springer, 2010.
3. M. Cornolti, P. Ferragina, and M. Ciaramita. A framework for benchmarking entity-annotation systems. In *22nd WWW*, pages 249–260, 2013.
4. S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, pages 708–716, 2007.
5. B. Ell, D. Vrandečic, and E. 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.

[10] See also `http://139.18.2.164/rusbeck/agdistis/supplementary.pdf` and `http://139.18.2.164/rusbeck/agdistis/appendix.pdf`

6. P. Ferragina and U. Scaiella. Fast and accurate annotation of short texts with wikipedia pages. *IEEE software*, 29(1), 2012.
7. J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, ACL '05, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.
8. D. Gerber, A.-C. Ngonga Ngomo, S. Hellmann, T. Soru, L. Bühmann, and R. Usbeck. Real-time rdf extraction from unstructured data streams. In *ISWC*, 2013.
9. J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. 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. J. Kleb and A. Abecker. Entity reference resolution via spreading activation on rdf-graphs. In *ESWC (1)*, pages 152–166, 2010.
11. J. Kleb and A. Abecker. Disambiguating entity references within an ontological model. In *WIMS*, page 22, 2011.
12. J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, Sept. 1999.
13. S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *15th ACM SIGKDD*, pages 457–466, 2009.
14. J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *SWJ*, 2014.
15. P. N. Mendes, M. Jakob, A. Garcia-Silva, and C. Bizer. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*, 2011.
16. R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *16th ACM Conference on information and knowledge management*, CIKM '07, pages 233–242, New York, NY, USA, 2007. ACM.
17. D. Milne and I. H. Witten. Learning to link with wikipedia. In *17th ACM CIKM*, pages 509–518, 2008.
18. A. Moro, A. Raganato, and R. Navigli. Entity linking meets word sense disambiguation: A unified approach. *TACL*, 2, 2014.
19. D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30:3–26, 2007.
20. L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *CoNLL*, 6 2009.
21. L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *ACL*, 2011.
22. M. Röder, R. Usbeck, S. Hellmann, D. Gerber, and A. Both. N3 - a collection of datasets for named entity recognition and disambiguation in the nlp interchange format. In *9th LREC*, 2014.
23. W. Shen, J. Wang, P. Luo, and M. Wang. Linden: linking named entities with knowledge base via semantic knowledge. In *21st WWW*, pages 449–458, 2012.
24. S. Singh, A. Subramanya, F. Pereira, and A. McCallum. Large-scale cross-document coreference using distributed inference and hierarchical models. In *49th ACL: Human Language Technologies-*, pages 793–803, 2011.
25. R. Speck and A.-C. N. Ngomo. Ensemble learning for named entity recognition. In *ISWC*, Lecture Notes in Computer Science. 2014.
26. E. F. Tjong Kim Sang and F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147, 2003.