Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

# Towards Knowledge-Driven Annotation

**Yassine Mrabet**
CRP Henri Tudor*
Luxembourg
yassine.mrabet@tudor.lu

**Claire Gardent**
CNRS/LORIA
Nancy, France
claire.gardent@loria.fr

**Muriel Foulonneau**
CRP Henri Tudor*
Luxembourg
muriel.foulonneau@tudor.lu

**Elena Simperl**
University of Southampton
Southampton, United Kingdom
e.simperl@soton.ac.uk

**Eric Ras**
CRP Henri Tudor*
Luxembourg
eric.ras@tudor.lu

**Abstract**

While the Web of data is attracting increasing interest and rapidly growing in size, the major support of information on the surface Web are still multimedia documents. Semantic annotation of texts is one of the main processes that are intended to facilitate meaning-based information exchange between computational agents. However, such annotation faces several challenges such as the heterogeneity of natural language expressions, the heterogeneity of documents structure and context dependencies. While a broad range of annotation approaches rely mainly or partly on the target textual context to disambiguate the extracted entities, in this paper we present an approach that relies mainly on formalized-knowledge expressed in RDF datasets to categorize and disambiguate noun phrases. In the proposed method, we represent the reference knowledge bases as co-occurrence matrices and the disambiguation problem as a 0-1 Integer Linear Programming (ILP) problem. The proposed approach is unsupervised and can be ported to any RDF knowledge base. The system implementing this approach, called *KODA*, shows very promising results w.r.t. state-of-the-art annotation tools in cross-domain experimentations.

## Introduction

With the exponential growth of information and the continuous specialization of domain-related knowledge, automatic text annotation becomes more and more important for largely computerized tasks such as information retrieval, eLearning activities, question answering or knowledge acquisition methods. In the last decade, this topic has been addressed by numerous works and from different perspectives including Natural Language Processing (NLP) (Yates et al. 2007; Popov et al. 2003), Databases (Gottlob et al. 2004; Venetis et al. 2011), and the Semantic Web (Cimiano, Ladwig, and Staab 2005; Suchanek, Sozio, and Weikum 2009; Mendes et al. 2011; Dill et al. 2003).

Text annotation can be considered as the association of text fragments to entities described in a more-or-less structured data space. Therefore, the richer the data are the more valuable the annotations are. This fundamental aspect explains the fast emergence of knowledge-based annotation systems. Today, with hundreds of Knowledge Bases (KBs) and more than 30 billion RDF[1] facts, linked open data[2] became indeed one of the richest data spaces to use for text annotation.

Most existing approaches focused on the performance of text annotation w.r.t. a specific (domain-related) KB. One of the explored tracks consists in extracting references to RDF resources using patterns defined by (hand written) regular expressions augmented with semantic tags (Cimiano, Ladwig, and Staab 2005; Popov et al. 2003; Suchanek, Sozio, and Weikum 2009). A second more scalable track consists in using learning corpora (i) to compute link probabilities between textual mentions and RDF entities or (ii) to train classification models according to machine-learning algorithms. However, few KBs are linked to (big) textual corpora that could be used for training. This fact led a wide majority of research works to focus on DBpedia[3] as it allows using Wikipedia as a learning resource. However, using only DBpedia as a target KB restricts the possible range of disambiguation. For instance, 57% of the named-entities in the Text Analysis Conference (TAC) 2009 were found to refer to an entity that did not appear in Wikipedia (McNamee et al. 2009) and Freebase is known to have at least 9 times as many entities. Moreover, the semantic annotation of domain specific texts would arguably benefit from semantic annotation methods targeting a related, domain specific KB.

In this paper, we present a novel, unsupervised and KB-agnostic approach to text annotation which associates Noun Phrases (NP) to RDF resources. Our approach differs from previous works in three main ways. First, it is knowledge- rather than text- or pattern-driven. Instead of using distributional similarity or patterns as a basis for semantic annotation, our approach solely relies on the RDF data. As a result, it is corpus independent; avoids data sparsity issues and

*On the 1st of January 2015, CRP Henri Tudor and CRP Gabriel Lippmann will merge to form the Luxembourg Institute of Science & Technology (http://www.list.lu)



[1] http://www.w3.org/TR/REC-rdf-syntax/

[2] http://www.linkeddata.org

[3] http://www.dbpedia.org

eschews the restrictions imposed by hand written patterns (the range of possible annotations is determined by the KB rather than by the patterns). Second, the approach is unsupervised in that it requires neither manual annotation of text with semantic data nor the specification of patterns. This allows for an approach which is KB-agnostic and fully automatic. Third, the semantic annotation is global rather than local in that all text segments are annotated simultaneously rather than individually.

One major issue when annotating text is disambiguation i.e., how to choose, given a set of possible semantic annotations, the correct one in the given context. In our approach, disambiguation is driven by "KB coherence" and semantic annotations are chosen to maximise graph connectedness in the KB. The intuition is that documents are semantically coherent and that preferring the RDF data which are most closely related in the KB mimics the contextual disambiguation provided by the surrounding text.

The paper is structured as follows. We start by situating our work with respect to previous approaches to semantic annotation. We then present our approach and compare the results obtained on 5 different benchmarks with those obtained by state-of-the-art systems. We conclude with pointers for further research.

## Related Works

Several existing methods in information extraction have been reused and adapted for RDF-based annotation. Named entity recognition methods using statistical machine learning and text patterns have been extended with categories defined as classes and instances of domain KBs, e.g. (Popov et al. 2003; Cimiano, Ladwig, and Staab 2005). Several annotation approaches focused on small KBs or on selected subsets of the KB concepts and entities and they did not consider annotating with unrestricted KB elements, especially for large cross-domain KBs such as DBpedia, YAGO or Freebase. In the scope of this paper we are primarily interested in unrestricted annotation w.r.t. large RDF datasets, for detailed surveys of different semantic annotation tools and methods, interested readers are referred to (Gangemi 2013) and (Reeve 2005).

Several approaches and tools addressed the challenge of annotating texts with reference to large (RDF) KBs. For instance, SOFIE (Suchanek, Sozio, and Weikum 2009) allows extending an RDF KB by extracting information from natural language texts. The system proposes a unified approach to pattern selection (relation extraction), entity disambiguation and consistency checking. It represents extraction hypotheses as clauses and solves them with a MAX-SAT solver that maximizes the weight of all the clauses and satisfies the consistency properties (e.g. functionality of properties, user-added domain rules). However, as noted by the authors, the approach is less effective if applied to small batches. This is mainly due to the fact that the solver is forced to annotate all entities and relations, which leads inevitably to wrong annotations if the correct alternative is not available in the documents or in the KB. DBpedia Spotlight (Mendes et al. 2011) is one of the state-of-the-art approaches for the semantic annotation of textual documents with DBpedia and is widely used in the Semantic Web community. It is based on a vector-space model and associates a "bag-of-words" to each DBpedia resource. The system ranks the DBpedia resources with a cosine similarity measure that is used to compare the vectors associated to DBpedia resources with the vectors representing the textual context of the target textual forms, called Surface Forms (SF) or mentions. In this setting, the semantic relatedness of the resources is not used, except for the "random" implicit links represented through the word-based vectors.

Other approaches considered directly Wikipedia as KB and addressed the task from a "wikification" point of view, i.e. linking the mentions to Wikipedia articles (Han, Sun, and Zhao 2011; Kulkarni et al. 2009; Milne and Witten 2013; Ferragina and Scaiella 2010; Milne and Witten 2008; Mihalcea and Csomai 2007; Ratinov et al. 2011; Cheng and Roth 2013). Existing wikification systems use the prior mention/title or mention/entity probabilities derived from Wikipedia hyperlinks. This probability showed to provide a strong baseline ranging from 70% to 80% in $F_1$ score. Different techniques are then used on top of this prior disambiguation information such as word-vector-based similarities, machine learning with SVM classifiers or the maximization of the relatedness between candidate Wikipedia articles using the hyperlinks that express often broad "*see also*" semantics.

In contrast to these approaches that rely extensively on the availability of large corpora such as Wikipedia we propose a fully-unsupervised approach that relies only on RDF data and can be ported to KBs that have no linked textual corpora. This approach, called Knowledge-Driven Annotation (*KODA*), does not rely on prior mention/entity probabilities or on similarity features extracted from manually annotated texts, nor does it use machine learning techniques. As input, the proposed approach needs only the KB facts and a set of lexical representation(s) of the KB entities. From a process point of view, we use the concept of global coherence between entities as in several related works cited above. However, we do not rely on wikipedia hyperlinks as a source of relatedness, but only on KB facts expressing explicit semantics. More precisely, we consider two entities as co-occurring in the KB if they are subject and object of the same RDF triple. As the number of properties and triples in a KB is far from being holistic w.r.t. real-world knowledge, we consider only $\{0, 1\}$ co-occurrence values. Therefore, this second difference with existing approaches can be summarized as statistical vs. semantic relatedness or as statistically-enhanced vs. purely-semantic relatedness.

## Knowledge-Driven Annotation (*KODA*)

### Overview

*KODA* consists in (1) a *retrieval module*, (2) a *disambiguation module* and (3) a *contextual search module* that are executed in sequence for each textual context to be annotated.

In the first **Retrieval module**, all Noun Phrases (NPs) are extracted from the target text and submitted as keyword queries to a search engine in order to retrieve a list of candidate RDF resources. This is made possible by an offline

indexation of the RDF resources using their Lexical Representations (LRs) (e.g. values of rdfs:label, foaf:name). This indexation is performed by considering each RDF resource as a document that has a URI field and the LRs as textual content.

For a given NP, the search engine will return a list of RDF resources ranked according to the TF-IDF score of their LRs. In order to obtain a fine-grained selection of the NPs in a given sentence, we select only one NP for each branch of its syntactic-parse tree using the TF-IDF scores; i.e. the NP that has the best TF-IDF score is preferred to all its descendant and ancestor NPs. Leaf NPs are also subdivided in different spans that are integrated into the parse tree as "virtual" NPs which can then be selected as potential annotation targets. In the remainder of the paper we will refer to all potential annotation targets as Surface Forms (SFs).

In the second **Disambiguation module**, *KODA* aims to select only one RDF resource for each SF. As we do not use learning material we built a *divide-and-conquer approach* where the Highly Ambiguous (HA) SFs are processed after the less Ambiguous (A) ones. This distinction is motivated by the fact that highly ambiguous forms have a lot of heterogeneous candidate resources which may flaw the disambiguation process by retrieving correct relationships between noisy/false candidates. The classification of SFs into { *A*, *HA* or *NA* (Non Ambiguous)} is based on the ratio of resources that share the maximal TF-IDF score according to a fixed threshold. For instance, for a threshold of 20, a ratio of 0.8 means that SFs that have more than 16 resources with maximal score are classified as *HA*. To better show the intuitions, let us consider the annotation of the following sentences with DBpedia:

*"Bean was selected by NASA as part of Astronaut Group 3 in 1963. A space was opened for him on the back-up crew for Apollo 9 when fellow astronaut Clifton Williams was killed in an air crash."*

Table 1 shows a subset of the SFs selected by our retrieval algorithm and information on the initial candidate resources for each SF. The high ambiguity threshold ratio has been fixed to 16/20 for this example.

In this disambiguation module, *KODA* processes only the SFs classified as *A* using a *co-occurrence maximization* process. Disambiguated SFs will then be moved to the *NA* class, SFs that are still in the *A* class after this first run will be moved to the *HA* set. For instance, in the example above, the SFs 2, 4 and 5 will be disambiguated and moved to the *NA* class. This is made possible with the link information provided by the following DBpedia triples:

| | | |
|---|---|---|
| *dbpedia*:*Apollo_9* | dbprop:operator | *dbpedia*:*NASA* |
| *dbpedia*:*Clifton_Willams* | db-owl:type | *dbpedia*:*NASA* |
| *dbpedia*:*Clifton_Willams* | rdf:type | *db-owl*:*Astronaut* |

It the third **Contextual Search module**, *KODA* tries to disambiguate the SFs in *HA*, using the SFs in *NA* as unambiguous references and a *contextual search* process. Contextual search consists in searching for candidates in the context of the previously selected RDF resources. For instance, searching the SF "Bean" in the KB context of *NA* = {*db*:*NASA_Astronaut_Group_3, db*:*Apollo_9, db*:*NASA, db*:*Astronaut, db*:*Clifton_Willams*} leads to only one candidate, i.e. *db*:*Alan_Bean*, which is the correct disambiguation, while the TF-IDF search for "Bean" led to more than 40 candidates with the same maximal score and ranked the correct disambiguation $49^{th}$. When several candidates are retrieved by contextual search, they are ranked according to their word-overlap with the SF, in that case disambiguation succeeds if only one candidate has the maximum word-overlap score.

Contextual search is the third and last process in *KODA*. The following section describes in more detail the inner co-occurrence maximization process of the second disambiguation module.

## Co-occurrence Maximization

Two RDF resources are considered as co-occurrent if they appear as subject and object in the same RDF triple. For an efficient computation, we transform the reference RDF KB into a 0-1 co-occurrence matrix in an offline process. The problem resolution is then translated to a 0-1 ILP problem. Basically, this disambiguation problem can be seen as a derivation of the Knapsack problem. However, reasoning on the item (resource) level will lead to a non-linear problem as we want to maximize co-occurrences. In order to have a linear representation we define the problem as the selection of co-occurring pairs of resources. This leads to a straightforward objective function to maximize but requires more elaborated constraints to obtain an exact representation of the original problem. For a set of target SFs: $F = SF_1, SF_2, .., SF_n$, a set of respective candidate resources $R = \{\{r_{11}, r_{12}, .., r_{s_1}\}, \{r_{21}, r_{22}, .., r_{s_2}\}, \ldots, \{r_{n1}, r_{n2}, .., r_{s_n}\}$, the objective function that needs to be maximized can be represented as:

$$Z = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \sum_{k=1}^{s_i} \sum_{l=1}^{s_j} w(r_{ik}) \times w(r_{jl}) \times c_{ik,jl} X_{ik,jl} \quad (1)$$

Where $w(r_{ik})$ is the TF-IDF score of $r_{ik}$ for $SF_i$, $c_{ik,jl} \in \{0, 1\}$ is the co-occurrence value of the resource pair ($r_{ik}$, $r_{jl}$) and $X_{ik,jl}$ is a boolean variable indicating whether this pair of resources is selected for the maximum (optimal solution) or not. This objective function must be subject to constraints that guarantee that only one candidate resource will be selected for each $SF \in F$. We define these constraints with the following semantics:

- A candidate resource $r_{ik}$ can be selected for a surface form $SF_i$ iff a co-occurring resource, $r_{jl}$, from another surface form $SF_j$, $i \neq j$ is selected. This is directly translated by the variable definitions. $X_{ik,jl} = 1$ implies that $r_{ik}$ is selected for $SF_i$ and $r_{jl}$ is selected for $SF_j$.

$$selected(r_{ik}, SF_i) \leftrightarrow \exists j, l \; s.t. \; X_{ik,jl} = 1 \vee X_{jl,ik} = 1 \quad (2)$$

- For a given pair of surface forms ($SF_i$, $SF_k$), at most one

| id | SF | Correct Disambiguation (CD) | Results Nb. | MS | rank of CD | Class |
|---|---|---|---|---|---|---|
| 1 | Bean | *dbpedia:Alan_Bean* | 1127 | $> 40$ | $> 40$ | *HA* |
| 2 | NASA | *dbpedia:NASA* | 456 | 14 | 6 | *A* |
| 3 | Astronaut Group 3 | *dbpedia:NASA_Astronaut_Group_3* | 440 308 | 1 | 1 | *NA* |
| 4 | astronaut | *db-owl:Astronaut* | 230 | 13 | 12 | *A* |
| 5 | Clifton Williams | *dbpedia:Clifton_Williams* | 11 491 | 3 | 3 | *A* |
| 6 | Apollo 9 | *dbpedia:Apollo_9* | 239 980 | 1 | 1 | *NA* |

Table 1: Example annotation scenario from our experiments: Results of the retrieval module (MS: number of results with the maximal score, CD: Correct Disambiguation)

resource pair ($r_{ik}$, $r_{jl}$) must be selected.

$$\forall i, j \sum_{k=1}^{s_i} \sum_{l=1}^{s_j} X_{ik,jl} \leq 1 \quad (3)$$

- If a resource $r_{ik}$ is selected for a surface form $SF_i$, (i.e. $\exists SF_j$ and $r_{jm}$ $s.t.$ $X_{ik,jm} = 1$) then $r_{ik}$ must be selected as well for all other *selected* resource pairs involving $SF_i$, represented by $\forall x, p, X_{ik,xp}$.

$$\forall (i,k,j,l) \; s.t. \; c_{ik,jl} \neq 0$$
$$\exists (x,z) \neq (j,l) \; s.t. \; c_{ik,xz} \neq 0 \rightarrow \sum_{l=1}^{s_j} X_{ik,jl} = \sum_{p=1}^{s_x} X_{ik,xp} \quad (4)$$

Constraint (2) will be used in the problem translation and result interpretation. Constraints (3) and (4) are the concrete constraints of the maximization problem to be solved. This $0-1$ Integer Linear Programming (ILP) problem is satisfiable; besides the trivial 0 solution, another trivial solution is to select a random non-zero co-occurrence between a random pair of SFs (if it exists) and to set all the other co-occurrence variables to 0.

**Theorem 1.** Constraints (2), (3) and (4) guarantee that for any solution, each candidate surface form $SF_i$ will have at maximum one selected candidate resource $r_{ik}$.

Proof for theorem 1 is provided in the extended version of the paper[4]. Several optimal solutions are possible according to the problem instance. In such cases, all optimal solutions are obtained by iterating the problem solving with additional constraints that prune the previously found solutions. The disambiguation is considered to be successful for a given SF if only one RDF resource is selected by the maximization process.

## Evaluation

As most of the benchmarks in RDF-based text annotation are based on DBpedia, we first implement and test the efficiency of *KODA*[5] on DBpedia in order to have relevant comparison with state-of-the-art systems. The Stanford CoreNLP API[6] has been used to obtain the NPs and the syntactic-parse trees. SolR[7] has been used as a search engine for the implementation of the retrieval module.

### Construction of the co-occurrence matrix

We studied the feasibility of the matrix-based representation empirically with DBpedia. The RDF triples were obtained from the latest available DBpedia dump (version 3.9). We considered 3 main files of the dump which contain the instance types, the mapping-based cleaned properties and the raw infobox properties. As expected, the density factor (i.e. the proportion of non zeros in the matrix) is very low: 2.67 $10^{-6}$. Starting from a theoretic number of co-occurrences of approximately $1,67\ 10^{13}$ we found a real number of non-zero co-occurrences of $4.4\ 10^{7}$. The construction of the DBpedia matrix lasted 5.4 hours with relevant partitioning and optimizations of a DBMS (experimental configuration: MySQL InnoDB with 8 Gb RAM and 4-cores laptop).

### Resources and configuration

We used the DBpedia RDFS labels, the FOAF names and nicknames from the cleaned-properties file[8] and the lexicalization dataset provided by the DBpedia spotlight team[9]. In order to have a coherent input w.r.t. our approach we had to process DBpedia redirections and disambiguation pages. These entities/pages do not correspond to "real-world" entities and do not have associated domain facts. Consequently, DBpedia resources representing disambiguation pages were removed and wiki-redirections were interpreted by adding the redirections' labels as additional lexicalizations to the "real" RDF resources.

*KODA*'s results were obtained by setting the number of first items (resources) returned by SolR queries to 40 for all SFs and the *HA* threshold to 9/20. We used the 0-1 ILP solver LpSolve[10] as it provided both acceptable performance and accessible implementation. Contexts of more than 10 SFs are automatically split into several independent contexts that preserve sentences segmentation. In our experiments,

[4]http://smartdocs.tudor.lu/koda/aaai-15-ext.pdf

[5]Online demonstration: http://smartdocs.tudor.lu/koda

[6]http://nlp.stanford.edu/software/corenlp.shtml

[7]http://lucene.apache.org/solr/

[8]http://wiki.dbpedia.org/Datasets

[9]http://wiki.dbpedia.org/Lexicalizations

[10]http://lpsolve.sourceforge.net/5.5/

the execution time varied from 16 seconds to 173 seconds. The most time-consuming task was syntactic parsing, especially for long sentences. The co-occurrence maximization cost varies according to the number of SFs in the textual context and to the number of candidates of each SF, but it did not exceed 10 seconds in the worst case as it processes subsets of $\approx 10$ textual mentions in different threads.

## Benchmarks and experimental settings

We performed three experimentations. In the first one we used 3 standard benchmarks from the literature and compared to DBpedia Spotlight and TagMe2 *which was found to have the best results* according to (Cornolti, Ferragina, and Ciaramita 2013). The ACQUAINT benchmark (724 SFs) (Milne and Witten 2008) consists of 50 short documents (25-300 words) extracted from the ACQUAINT text corpus, a collection of newswire stories. The MSNBC data (660 SFs)(Cucerzan 2007) includes the top two stories in the ten MSNBC news categories. In both benchmarks SFs were disambiguated by applying a classifier and manually verifying the detected links. In these wikification benchmarks only "link-worthy" SFs were annotated. Therefore, to assess the coverage of our approach we test it on the IITB corpus proposed in (Kulkarni et al. 2009). The IITB corpus contains more than 19K manual annotations for 103 documents.

In a second experimentation we compare *KODA* to 8 annotation systems using their online demonstrations. For that purpose, we manually built a benchmark of 404 annotated surface forms, called WikiNews. The construction of this benchmark was motivated by the fact that we want a reliable evaluation of the coverage of our approach (which is not possible with ACQUAINT and MSNBC) and an easily-processable corpus both in terms of portability and execution time (which is not possible with the IITB corpus). The final gold annotations of the WikiNews corpus were produced by annotating manually (i) 5 news documents from CNN online, published on April 14 2014, and talking about unrelated topics and (ii) 5 documents having the same topic (i.e. Astronauts) collected from Wikipedia abstracts. One annotator performed the manual annotations. The pre-established guideline was to annotate the minimum number of trigger words that were sufficient to refer to the DBpedia resource and to not resolve implicit co-references (e.g. pronouns, hypernyms).

In a third experimentation, we tested the portability of *KODA* to other (domain-specific) KBs using the NCI[11] and Snomed-CT[12] taxonomies. These two taxonomies do not contain instances but only RDFS/OWL classes and property definitions. The direct co-occurrences are therefore obtained only from *rdfs*:*subClassOf* relations. To obtain more co-occurrences we also considered the concepts that are domain and range of the same property as co-occurrent. As no standard benchmarks are available for these KBs, we constructed a reference corpus, $PubMed_1$, consisting of 312 annotated SFs obtained from 5 PubMed articles about Amygdala, Influenza and Ebola. All corpora used in the experiments as well as the results obtained by *KODA* are available on the project website[13].

## Results and discussion

*KODA* achieved the highest F1 score in the first experimentation and outperformed significantly the state-of-the-art systems (cf. table 2). This first observation supports our main assumption, i.e. that *unsupervised methods relying only on KB content can achieve efficient annotation without using supplementary corpus-based statistics or corpus-based classification*.

DBpedia Spotlight does not use KB-level coherence and relies mainly on word vectors associated to the RDF resources. Therefore *these results show that global coherence alone is able to significantly outperform learned word vectors for the disambiguation task*. TagMe2 uses Wikipedia's hyperlink anchors (<a> tags) as a predefined list of "recognizable" noun phrases and uses internal Wikipedia hyperlinks to perform a global coherence-based disambiguation. Hence, TagMe2 fails in reaching a high coverage due to its entity detection method based on exact matches according to existing anchors. *KODA* obtained the best precision on the three benchmarks; we explain this mainly by its divide-and-conquer approach that addresses the highly ambiguous SFs in the end, when more references have been produced by the disambiguation of the other, less ambiguous forms.

An important performance decrease can be noticed between the IITB corpus and the MSNBC and AQUAINT corpora. This is mainly due to the high heterogeneity of the annotated SFs in IITB which contain many secondary forms such as years and numbers that cannot be disambiguated correctly using the KB-level coherence; they actually add significant noise for coherence-based disambiguation. This observation is supported by the results of TagMe2, which also uses global coherence, and which also had its worst performance on IITB, while the Spotlight system which relies on corpus-based word vectors had actually its best performance on IITB due to its local disambiguation (i.e. each surface form is disambiguated separately).

Many other annotation tools use DBpedia and/or Wikipedia as a background knowledge base, several of them have online demonstrations. In table 3 we present the results obtained on the WikiNews corpus for *KODA* and 8 other systems available online: DBpedia spotlight[14], AIDA[15], Wikipedia Miner[16], TagMe2[17], Calais[18], Cicero[19], FOX[20], and AlchemyAPI[21]. The state-of-the-art systems were manually evaluated w.r.t. the benchmark using their online interfaces. Mentions that have an overlap with a reference mention from the gold set were considered as cor-

[11] http://evs.nci.nih.gov/

[12] http://bioportal.bioontology.org/ontologies/SNOMEDCT

[13] http://smartdocs.tudor.lu/koda/datasets.html

[14] http://dbpedia-spotlight.github.io/demo/

[15] https://gate.d5.mpi-inf.mpg.de/webaida/

[16] http://wikipedia-miner.cms.waikato.ac.nz/demos

[17] http://tagme.di.unipi.it/

[18] http://viewer.opencalais.com/

[19] http://demo.languagecomputer.com/cicerolite/

[20] http://139.18.2.164:4444/demo/index.html#!/demo

[21] http://www.alchemyapi.com/

| System | AQUAINT | | | MSNBC | | | IITB | | |
|---|---|---|---|---|---|---|---|---|---|
| | Precision | Recall | $F_1$ | Precision | Recall | $F_1$ | Precision | Recall | $F_1$ |
| Spotlight | 17.8 | 48.0 | 26 | 31.7 | 34.7 | 33.1 | 43.4 | 48.9 | 46.0 |
| TagMe2 | 41.2 | 51.4 | 45.7 | 43.1 | 50.8 | 46.6 | 41.6 | 40.0 | 40.8 |
| KODA | **84.95** | **68.64** | **75.93** | **90.84** | **78.18** | **84.03** | **82.30** | **54.01** | **65.2** |

Table 2: Precision, Recall and $F_1$ scores on AQUAINT (724 SFs), MSNBC (660 SFs) and IITB (19K SFs)

| | **KODA** | **Spotlight** | **AIDA** | **WikiMiner** | **TagMe2** | **Calais** | **Cicero** | **FOX** | **Alchemy** |
|---|---|---|---|---|---|---|---|---|---|
| Precision | **81.84**% | 91.91% | 65.75% | **80.84**% | 58.42% | 87.66% | 91.79% | 84.09% | 77.27% |
| Recall | **59.15**% | 30.94% | 23.76% | **61.63**% | 66.08% | 33.41% | 44.30% | 27.47% | 25.24% |
| F1 | **68.66**% | 46.29% | 34.90% | **69.93**% | 62.01% | 48.38% | 59.75% | 41.41% | 38.05% |

Table 3: Evaluation on WikiNews (404 SFs)

rect if their semantic annotation (associated RDF resource) is correct. For Calais, as the annotation does not refer directly to DBpedia entities, we converted the annotations using the DBpedia naming procedure[22] and considering the annotated SF as a Wikipedia title. For the other systems, we considered only DBpedia/Wikipedia-based annotations. When a precision/recall knob was available, we selected the maximum-recall parametrization, which corresponds to one of the challenges addressed by *KODA* (i.e. aggressive annotation of unstructured texts). This choice is supported by the results which show that *KODA* has a better recall (up to 2 times better) than 6 systems.

*KODA*'s $F_1$ score is second (68.66% $F_1$) when compared to the other 8 systems (cf. table 3). Wikipedia Miner performs slightly better with an $F_1$ score of 69.93%. However, its underlying machine-learned semantic relatedness can not be ported to other knowledge bases if no learning corpus is available. AIDA relies on the Stanford parser to detect named entities; this limits the recall w.r.t. approaches that use the knowledge bases to select the target mentions. *KODA* achieved 59.15% recall with its method using TF-IDF scores to select the best candidate mention for each branch of the syntactic parse trees. TagMe2 also achieved a high recall (66.08%), however, it failed in achieving a high precision (58.42%).

Hence, the observation from the first experimentation is sustained in the second; *the divide-and-conquer approach of KODA made a significant difference in precision w.r.t. systems that use similar global-coherence aspects*. Also, the co-occurrence maximization proposed in *KODA* does not allow to select resources that have no connections (only co-occurring pairs are selected) while existing approaches use the connectedness only as an additional scoring feature.

An analysis of the incorrect annotations produced by *KODA* showed that they are mainly due to the statistical behaviour of the maximization which may force the annotation of some SFs in order to optimize the objective function. On average *KODA* achieved an average F-measure of 73.45% on 4 different open-domain benchmarks with a fully unsupervised approach that relies only on the KB content and uses no textual patterns, no learning corpora and no prior disambiguation information.

[22]http://wiki.dbpedia.org/Datasets#h434-3

| | Snomed-CT | NCI | DBpedia |
|---|---|---|---|
| Precision | **83.8**% | 82.44% | 76.37% |
| Recall | 38.14% | 34.61% | **44.55**% |
| $F_1$ | 52.42% | 48.75% | **56.27**% |

Table 4: *KODA* results on $PubMed_1$ (312 SFs) with 3 KBs

In a first portability test, we deployed *KODA* with domain-specific taxonomies: i.e. Snomed-CT and NCI and tested its performance on the $PubMed_1$ corpus. As *KODA* is KB-agnostic, its application to these biomedical taxonomies did not require any implementation change. The obtained performance however depends on the richness of the reference KBs in terms of relationships and in terms of lexical representations.

This aspect is highlighted when we compare the results obtained by these two KBs with the results obtained by DBpedia (cf. table 4). While both domain-related taxonomies obtained the best precision, DBpedia performed better in recall, mainly due to its size and to the number of available lexical representations for each RDF resource. Taxonomies are not the best competitors to populated KBs such as DBpedia, however, this first portability study shows that the approach is still efficient in terms of precision even when ported to domain-specific KBs that lack a rich set of relationships. In this experiment these few co-occurrences still allowed to retrieve 30.9% of the correct annotations for NCI and 30.3% of the correct annotations for SNOMED.

Another important aspect of text annotation with RDF KBs is the adequacy of the KB with the annotated text. In coming work we plan to derive automatically a score to characterize this adequacy according to the ambiguity level, the number of retrieved RDF resources and the number of KB-

level relations.

## Conclusion

We presented an unsupervised approach for the semantic annotation of texts with RDF Knowledge Bases (KB). The proposed approach, called *KODA*, allows projecting large KBs in textual contexts. *KODA* exploits the adjacency matrices of RDF KBs to select the RDF resources that maximize their pairwise co-occurrences. We represented the disambiguation problem as a 0-1 Integer Linear Programming problem that guarantees that each optimal solution will associate one RDF resource to each SF. The obtained results on several benchmarks are very promising and open a new solution track for text annotation with KBs that lack learning corpora. Coming work will include multi-lingual text annotation and integration of additional (open-)domain KBs.

## Acknowledgments

This work was carried out during tenure of an ERCIM "Alain Bensoussan" Fellowship Programme grant, supported by the National Research Fund (Luxembourg) and by the Marie Curie Co-funding of Regional, National and International Programmes (COFUND) of the European Commission.

## References

Cheng, X., and Roth, D. 2013. Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013*, 1787–1796.

Cimiano, P.; Ladwig, G.; and Staab, S. 2005. Gimme' the context: Context-driven automatic semantic annotation with c-pankow. In *Proceedings of the 14th International Conference on World Wide Web*, 332–341. ACM.

Cornolti, M.; Ferragina, P.; and Ciaramita, M. 2013. A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd international conference on World Wide Web*, 249–260.

Cucerzan, S. 2007. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, volume 7, 708–716.

Dill, S.; Eiron, N.; Gibson, D.; Gruhl, D.; Guha, R.; Jhingran, A.; Kanungo, T.; Rajagopalan, S.; Tomkins, A.; Tomlin, J. A.; and Zien, J. Y. 2003. Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, 178–186. ACM.

Ferragina, P., and Scaiella, U. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, 1625–1628. ACM.

Gangemi, A. 2013. A comparison of knowledge extraction tools for the semantic web. In Cimiano, P.; Corcho, O.; Presutti, V.; Hollink, L.; and Rudolph, S., eds., *The Semantic Web: Semantics and Big Data*, volume 7882 of *LNCS*. Springer Berlin Heidelberg. 351–366.

Gottlob, G.; Koch, C.; Baumgartner, R.; Herzog, M.; and Flesca, S. 2004. The lixto data extraction project: Back and forth between theory and practice. In *Proceedings of the Twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '04, 1–12.

Han, X.; Sun, L.; and Zhao, J. 2011. Collective entity linking in web text: A graph-based method. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, 765–774. ACM.

Kulkarni, S.; Singh, A.; Ramakrishnan, G.; and Chakrabarti, S. 2009. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 457–466. ACM.

McNamee, P.; Dredze, M.; Gerber, A.; Garera, N.; Finin, T.; Mayfield, J.; Piatko, C.; Rao, D.; Yarowsky, D.; and Dreyer, M. 2009. Hltcoe approaches to knowledge base population at tac 2009. In *Text Analysis Conference (TAC)*.

Mendes, P. N.; Jakob, M.; García-Silva, A.; and Bizer, C. 2011. Dbpedia spotlight: Shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, 1–8. ACM.

Mihalcea, R., and Csomai, A. 2007. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of CIKM*, CIKM '07, 233–242. ACM.

Milne, D., and Witten, I. H. 2008. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, 509–518. ACM.

Milne, D., and Witten, I. H. 2013. An open-source toolkit for mining wikipedia. *Artif. Intell.* 194:222–239.

Popov, B.; Kiryakov, A.; Kirilov, A.; Manov, D.; Ognyanoff, D.; and Goranov, M. 2003. Kim: Semantic annotation platform. In Fensel, D.; Sycara, K.; and Mylopoulos, J., eds., *The Semantic Web - ISWC 2003*, volume 2870 of *LNCS*. Springer. 834–849.

Ratinov, L.; Roth, D.; Downey, D.; and Anderson, M. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 1375–1384. Association for Computational Liguistics.

Reeve, L. 2005. Survey of semantic annotation platforms. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, 1634–1638. ACM Press.

Suchanek, F. M.; Sozio, M.; and Weikum, G. 2009. Sofie: A self-organizing framework for information extraction. In *Proceedings of the 18th International Conference on World Wide Web*, 631–640.

Venetis, P.; Halevy, A.; Madhavan, J.; Paşca, M.; Shen, W.; Wu, F.; Miao, G.; and Wu, C. 2011. Recovering semantics of tables on the web. *Proc. VLDB Endow.* 4(9):528–538.

Yates, A.; Cafarella, M.; Banko, M.; Etzioni, O.; Broadhead, M.; and Soderland, S. 2007. Textrunner: Open information extraction on the web. In *NAACL-Demonstrations '07*, 25–26. Association for Computational Linguistics.