Reconnecting Digital Publications to the Web using their Spatial Information

Ben De Meester, Tom De Nies, Ruben Verborgh, Erik Mannens, and Rik Van de Walle
Ghent University - iMinds - MMLab
Gaston Crommenlaan 8 bus 201
9050, Ledeberg-Ghent, Belgium
{firstname.lastname}@ugent.be

ABSTRACT

Digital publications can be packaged and viewed via the Open Web Platform using the EPUB 3 format. Meanwhile, the increased amount of mobile clients and the advent of HTML5's Geolocation have opened a whole range of possibilities for digital publications to interact with their readers. However, EPUB 3 files often remain closed silos of information, no longer linked with the rest of the Web.

In this paper, we propose a solution to reconnect digital publication with the (Semantic) Web. We will also show how we can use that connection to improve contextualization for a user, specifically via spatial information. We enrich digital publications by connecting the detected concepts to their URIs on, e.g., DBpedia, and by devising an algorithm to approximate the location of any detected concept, we can provide a user with the spatial center of gravity of his reading position.

The evaluation of the location approximation algorithm showed a high recall, and the high correlation between estimation error and standard deviation can provide the user with a sense of correctness (or spread) of an approximation. This means relevant locations (and their possible radius) can be shown for a user, based on the content he or she is reading, and based on his or her location. This methodology can be used to reconnect digital publications with the online world, to entice readers, and ultimately, as a novel location-based recommendation technique.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based services*

Keywords

Approximation; DBpedia; EPUB 3; Geolocation; RDFa

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2741714.

1. INTRODUCTION

Since the fall of 2011, the *International Digital Publishing Forum* (IDPF)[1] finalized EPUB 3, the latest version of the open e-book standard [3]. The most important improvement of EPUB 3 in comparison with EPUB 2 is the support for the *Open Web Platform*, including HTML5, CSS3, and the JavaScript APIs. Although this means EPUB 3 has the potential to make digital publications a first-class citizen of the Web, the current situation is that a digital publication is packaged, and that its content is no longer linked with the rest of the Web. This means that content inside EPUB 3 publications remains closed off, and undiscoverable. Since its last update in June 2014, the EPUB 3 format has become an ISO standard. But more importantly, since then, EPUB 3 files are allowed to be annotated with semantic annotations: either RDFa [5] or microdata [6]. These semantic annotations hold the power to reconnect digital publications, and make their content linkable and discoverable once more [9].

However, there are many ways to connect content with the (Semantic) Web. Since HTML5's support for Geolocation, a whole new array of possibilities are available for Web developers to create location-based recommendations. This has been successfully applied in marketing and advertising [2], but always on already structured and well annotated resources. As digital publications are usually read on mobile devices, and as these devices support Geolocation, location-based connections could be the means to remove the barriers between digital publications and the online Web.

This paper's contribution is laying the ground work towards connecting a user's context with digital publications using the Semantic Web (in this paper, targeted specifically to the user's location). As such, its contribution is twofold: documenting a way of connecting digital publications with the Semantic Web, and evaluating a way of connecting the Semantic Web with the user's context. This first contribution is handled by using Natural Language Processing techniques to connect unstructured legacy content from Project Gütenberg[2] with links on the Semantic Web (in this paper specifically with DBpedia entities). The second contribution is handled by devising a way to connect the user's location with the location of the entities as mentioned on the currently read page. This second contribution involves estimating locations of entities that inherently do not have a location. The work currently described enables enhanced contextualization for the reader (i.e., knowing what the location gravity point is of the content the user is currently

[1] `http://idpf.org/`
[2] `http://www.gutenberg.org/`

reading). However, future work will indicate how this work will be used to enable the discovery of relevant publications based on the location of the user and the estimated spatial center of gravity of digital publications.

After reviewing relevant work in Section 2, we explain the methodology of reconnecting digital publications with the Web and the user's location in Section 3. In Section 4, we elaborate on how we estimate the location of DBpedia entities, which we thereafter evaluate. The end result is shown in Section 6, and finally, we conclude in Section 7.

2. RELATED WORK

As our research goal is to connect digital publications with their readers and the rest of the Web using mentioned locations in the publication, we will review the related work on publishing linkable content (Subsection 2.1), retrieving mentioned entities (Subsection 2.2), and geospatial data on the Web (Subsection 2.3), respectively.

2.1 Publishing Linkable Content

The original goal of the Semantic Web is to provide for machine-readable data. The first step to this end is to publish new data (or convert existing data) on the Web in a format that machines can understand naturally, which in turn would create a Web of data that can be processed by machines. Whereas the human-readable Web consists of links inside HTML, Linked Data uses RDF [7] to describe its data.

There are multiple ways of connecting this machine-understandable format to the current Web. On the one hand, content negotiation can be used to serve web pages as either (human-readable) HTML, or as a (machine-readable) serialization of RDF. On the other hand, RDFa and microdata are two distinct but similar ways of incorporating these semantic concepts and relations inside the HTML page, embedding the machine-understandable concepts within the human-readable format.

To query Linked Data, SPARQL is used as the de facto query language, the most widely known SPARQL endpoint being DBpedia [1], an automated Linked Data conversion from the data of – among others – Wikipedia. Triple Pattern Fragments [15] is a more affordable way of implementing a database endpoint: instead of deferring the heavy processing of every query to the server, a lightweight server is implemented that only returns triple patterns, and the SPARQL query is resolved on the client. This way, it becomes possible to create Linked Data applications on top of the actual servers that host the data, instead of needing to install a personal DBpedia instance, separate from the primary server.

2.2 Mentioned Entities

Concepts in a text are usually detected in three phases [10] (as indicated in Listing 1[3]). First, *Named Entity Recognition* (NER) is performed to detect entities in a text. Second, a *Named Entity Disambiguation* (NED) engine tries to classify these entities in, e.g., places, people, or objects. Third, these classified entities are (optionally) connected to their semantic URIs. Many disambiguation engines exist, such as AlchemyAPI[4] and OpenCalais[5], however, NERD [11], AGDISTIS [14] and DBpedia Spotlight [4] also connect the detected concepts to their URI on `http://dbpedia.org`. Also, the latter two are open-source, whereas the other engines are closed-source or commercial products. As in this work we do not focus only on geographical names, we will use more generic NER engines than only geographic disambiguation engines such as documented in, e.g., [12].

```
Barack Obama
is the president
of the USA
→ NER →
[Barack Obama]
is the president
of the [USA]
→ NED →
[PERSON|Barack Obama]
is the president
of the [PLACE|USA]
→ Connecting the concepts with their URIs →
[dbr:Barack_Obama|Barack Obama]
is the president
of the [dbr:United_States|USA]
```

Listing 1: The three steps of entity linking

2.3 Geospatial data on the Web

A lot of structured geospatial datasets are available on the Web, the biggest ones being OpenStreetMap[6] and GeoNames[7]. Google Maps[8] is a popular geographic Web service that allows only a very restricted querying API, targeted for manual use. These services commonly use a gazetteer (a geographical dictionary) to look up coordinates based on a given name. The drawback is that, e.g., Paris (Texas) and Paris (France) have the same given name, and thus a manual intervention is needed to select the correct coordinates. This is a consequence of the fact that these datasets are available on the Web using their (proprietary) Web service API, and not via RDF. LinkedGeoData [13] is an effort to convert the data of OpenStreetMap to RDF. However, the connection between an entity in OpenStreetMap and DBpedia is not explicit in the online endpoint, making it hard to use this dataset to retrieve locations for DBpedia links.

Previous research efforts include georeferencing Web resources [8, 16], however, this work was targeted on entire articles, not on the individual concepts. To the best of our knowledge, this paper is the first attempt at deriving the spatial location of linked data entities that do not have spatial coordinates.

3. METHODOLOGY

Figure 1 shows the used methodology. At design time, a legacy publication is enriched by detecting its concepts `(1)`. When reading that enriched book, the concepts that appear on the current page are detected `(2)`, and their linked DBpedia content is fetched `(3)`. These concept's spatial coordinates are then derived `(4)`, and the reader's location is shown together with the detected locations `(5)`.

[3]From now on, we will use `dbr` as prefix for `http://dbpedia.org/resource/`

[4]`http://www.alchemyapi.com/`

[5]`http://www.opencalais.com/`

[6]`http://www.openstreetmap.org/`

[7]`http://www.geonames.org/`

[8]`https://www.google.be/maps`

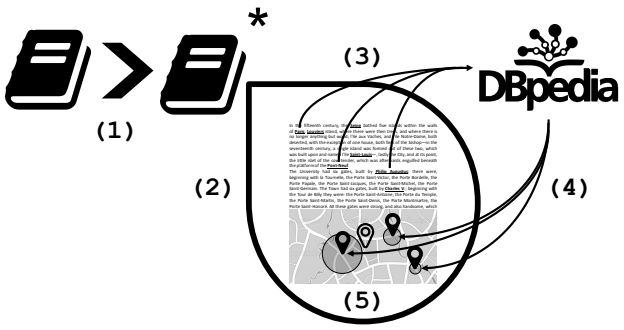

Figure 1: The detected DBpedia concepts inside a digital publication are used to retrieve the (estimated) location of the mentioned entities on an e-book page

This methodology materialized itself in the following technology stack:

- The EPUB 3 files are enriched with DBpedia Spotlight [4], to be able to detect semantic concepts while reading a digital publication.
- EPUB.js[9] is used as an open source e-reader.
- GreenTurtle[10] is a JavaScript library that implements the RDFa specification, and can be used to detect the RDFa annotations inside an HTML5 publication.
- The triple pattern fragments client is used to reliably query the fragments endpoint of DBpedia[11].
- The concept's spatial coordinates are derived or approximated using a custom algorithm (this approximation is elaborated on in Section 4).
- The reader's position is derived using HTML5's Geolocation.
- Leaflet.js[12] is used to visualize all map points close to the reader. Leaflet.js was chosen among alternatives for its maturity and open source nature.

The major contributions this paper presents are the enrichment of legacy EPUB 3 files, the adjustment of the EPUB 3 reader to catch the detected concepts on the currently viewed page, and the derivation of (approximated) coordinates based on DBpedia concepts.

A publication is enriched beforehand instead of enriching it at runtime, as EPUB 3 source files cannot be updated inside a reading environment, and HTML5 Web Storage recommends a five megabyte limit per origin, which we considered too low given the possible size of a digital publication. Being unable to update an EPUB 3 file in the e-reader means that semantic annotations cannot be persisted, and that the same annotation effort would need to be repeated every time the same page is opened in an e-reader.

[9] `https://github.com/futurepress/epub.js`
[10] `https://github.com/alexmilowski/green-turtle`
[11] `http://fragments.dbpedia.org/`
[12] `http://leafletjs.com/`

```
<p>Paris is the capital of France</p>
                    ⇒
<p>
    <span about="dbr:Paris">Paris</span>
    is the capital of
    <span about="dbr:France">France</span>
</p>
```

Listing 2: Connecting words to their semantic concepts using RDFa

To enrich them, we used DBpedia Spotlight[13] because it is open source, and can be installed locally instead of using its web service API. This local service will thus output more reliable and consistent results than its public service. However, enriching a publication via DBpedia Spotlight was not straightforward for two reasons: (1) DBpedia does not handle EPUB 3 out of the box, and (2) it is not built for analyzing large pieces of content. To this end, we had to unpack the EPUB 3 files to extract the text. Next, we chopped the text into smaller blocks (as DBpedia uses only a limited amount of contextualization, this will have little effect on the detection results). Each text was sent to a local DBpedia Spotlight instance and the original HTML was annotated accordingly with RDFa tags to connect the entities inside the HTML with their DBpedia links (see Listing 2 for an HTML+RDFa example). Finally, the unpacked EPUB 3 file with the annotated HTML files is repacked.

The resulting EPUB 3 file can be read by any EPUB 3 compliant e-reader (which is a functionality that more and more e-readers are providing recently[14]). However, to show a map depending on the viewed page, the publications should be able to receive updates when a page is flipped. As this functionality is not integrated in the EPUB 3 standard, it was necessary to either create a custom plugin for the EPUB 3 file to constantly check which page is visible, or to create a plugin for the e-reader, to update the location map every time a page is turned. The choice was made to adapt the e-reader to provide for a *geolocation plugin*. This last option is much less resource-intensive and has as result that an e-reader with such a plugin can provide this functionality for every e-book it loads. Otherwise, this geolocation functionality would need to be implemented in each e-book, and – within this e-book – for every e-reader, as the way of rendering could be different for different e-readers.

Every time a page is turned, the e-reader checks the visible content for possible RDFa tags. Every RDFa tag corresponds to a DBpedia entry, whether it is a place, a person, an organization, or something else. For every DBpedia entry, an exact or an approximated location with a certain radius is derived (further described in Section 4). These locations are then used to update the map inside the e-reader application to show relevant locations, based on the content that the user is reading. For every location, the radius is also visualized, giving a hint to the user of the possible estimation error. As the map will center on the current location of the user, only nearby locations will be shown. These results are contextualized for every reader based on the publication he or she is reading, and based on the reader's current location.

[13] Version 0.7, with the `en_2+2` dataset to train the engine.
[14] `http://epubtest.org/results/`

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX grs: <http://www.georss.org/georss/>
SELECT ?concept ?point
WHERE {
  dbr:Minnesota ?p ?concept.
  ?concept grs:point ?point
}
```

Listing 3: Getting coordinates of the linked concepts of Minnesota on DBpedia. Note how we do not specify how the entities are related to each other (the generic ?p in the query).

4. ADDING SPATIAL COORDINATES TO DBPEDIA ENTITIES

The following algorithm will approximate the location of entities that do not have coordinates inserted in DBpedia. These entities are typically not places, but rather people and events. In the following paragraphs however, we will focus on approximating the locations of actual places, to be able to evaluate the approximation. Two distinct methods are devised and evaluated, namely: (1) a link-based method, which looks for locations in the linked concepts of the root concept to approximate the location of the root concept, and (2) an abstract-based method, which looks for the locations of the concepts detected in the abstract of the root concept (i.e., the short description of that root concept) to approximate its location.

Link-based.

E.g., *Minnesota* has no geolocation in DBpedia, but it is linked to (among others) *Saint Paul* and *Minneapolis*, which do have coordinates. We query Dbpedia for these links (see Listing 3 for the SPARQL query), and return their average and standard deviation as an approximation to the coordinates of Minnesota[15]. There are two options, taking into account either all found links, or only the distinct links. Evaluation showed that taking into account all found links improved performance (see Section 5). When we average all found coordinates, we get N 45.8801 W 92.1542 as an approximation result for Minnesota. The euclidean distance with the actual location of Minnesota as found on Geonames.org – namely N 46.2502 W 94.2506 – is $167.33km$. As Minnesota comprises of an area of $225\,181km^2$, we consider this a fair approximation.

Abstract-based.

However, when no links with coordinates are available, we can still approximate the location of a DBpedia entity by using its abstract. E.g., the DBpedia entity *110 Livingston Street* does not have a coordinate, and does not have any links with coordinates in its DBpedia entry. But by disambiguating the entities mentioned in its abstract – using the same entity disambiguation as mentioned before for the enriching process of the EPUB 3 file – we do get some links for places such as *Manhattan* and *Brooklyn*. By using the same averaging technique, we get N 40.6956 W 73.9885. As this is a very specific location (i.e., a building), we also expect a more precise approximation, and indeed, the distance between the approximation and the location as found on OpenStreetMap is $0.52km$, a lot smaller than for the approximation of Minnesota. This approximation is again quite fair, using only the entities recognized in the abstract of the DBpedia resource.

[15]Note that, by averaging the found coordinates, we assume each found coordinate to be equally important. As we want this method to be generic for any kind of entity, we will not make any assumptions about the used coordinates. If we would target purely on actual locations, better techniques could be devised, e.g., a weighted average towards more populated regions

5. EVALUATION

Both methods can thus be used to approximate coordinates of places that do not have coordinates entered in their Wikipedia/DBpedia page. To evaluate these methods, we created a test set of locations by fetching the corresponding DBpedia links of a Wikipedia page with a lot of spatial objects[16]. Based solely on DBpedia, we derived the exact location, the location based on the detected concepts in the abstract, the location based on the linked locations, and the combinations of these methods (Table 1 shows the results for the 468 locations out of the 540 locations in the test set that had their coordinates set in DBpedia 2014)[17]. This way, we can evaluate the approximations by using their actual locations as inserted in DBpedia as a ground truth.

Table 1 shows the results of the three methods and the combination of the link-based and the abstract-based approach, on the one hand when the abstract-based approximation is calculated first, and if that fails, the result of the link-based approximation is calculated, and vice versa on the other hand. First, the amount of locations for which the method finds a location is given, absolute in the FREQUENCY column, and relative in the RECALL column, respectively. The average error is the average amount of kilometers between the found approximation and the coordinates as found in DBpedia. The F-MEASURE (best result in bold) is the harmonic mean between recall and precision (see Equation 1). In this case, we consider the inverted normalized average error as the precision of the approximation results, however, this is a precision metric relative to the results of this evaluation, and as such can only be used to compare these methods with each other, and not with other methods without adjusting the precision metric.

$$F = 2 \cdot \frac{recall \cdot precision}{recall + precision} \tag{1}$$

The table shows how the abstract-based approach can retrieve more results (i.e., a higher recall), but its average error is also significantly higher. When reviewing the F-measures, we see that the combination where first a link-based approximation is calculated returns the highest F-measure, thus, that is mathematically the best option when trying to optimize for both recall and the average error. When we calculate the correlation between the error and the standard deviation of the approximation, we note a significant positive correlation: 80.92% and 79.29% for the link-based and the abstract-based approach, respectively.

[16]http://en.wikipedia.org/wiki/List_of_buildings_and_structures

[17]The resulting data is available at http://users.ugent.be/~bjdmeest/locweb2015/data.json

Method	Frequency	Recall	Avg Error (km)	F-measure
Abstract-based approx.	457	97.65%	603.22	76.35%
Distinct link-based approx.	392	83.76%	408.99	87.89%
Link-based approx.	392	83.76%	378.11	91.16%
Combination (first abstract-based)	468	100.00%	601.24	77.22%
Combination (first link-based)	468	100.00%	444.97	**91.88%**

Table 1: The comparison between methods and combinations for detecting the locations of 468 DBpedia concepts. The table shows that first calculating the link-based approximation and falling back to the abstract-based approximation yields the best results.

6. RESULTS

First of all, the evaluation shows that a distinct link-based approach is less accurate than a link-based approach. For example, the *Roberto Clemente Coliseum* links many times to Puerto Rico, and once to Ljubljana, as a certain sporting event that was hosted in the Roberto Clemente Coliseum was hosted before in Ljubljana. When we only take into account the distinct links, the approximation is somewhere in between Puerto Rico and Ljubljana, which results in a big error. When we do not filter on distinct links, the approximation is weighted towards Puerto Rico, which is preferred.

The evaluation also shows that a combination of both approaches is the best option to achieve the most accurate results. As this evaluation has been performed on real DBpedia data, and as the proposed methods do not take into account that the entities are actual locations when calculating their approximations, we believe we can achieve similar results for DBpedia resources where no coordinates are present. A location can thus also be approximated for resources that are not actually locations. Taking into account the fact that the errors and standard deviation are highly correlated, we can actually also approximate the possible error of this approximation by using the standard deviation to define the radius of the approximation. High standard deviations will induce high radii, and thus span larger areas than more precise approximations.

The fact that the average error is high can be explained as follows. Firstly, the test set comprised of a variety of locations, from small landmarks to countries and nations. It is presumable that an approximation of the location of the United States of America will incorporate a bigger error and could still be more correct than an approximation of location of the White House with a very small error. And secondly, some approximations consisted of a very small amount of points (for the link-based approximation, the average of 57 out of the 392 found approximations consisted of only a single approximation point, and for the abstract-based coordinate this was true for 16 out of the 457 found approximations). Figure 2 shows the distribution of the average errors for both approximation methods, and illustrates how the link-based method generally returns more accurate results (e.g., 32% of the link-based approximations have an error of order $10km$ or less, compared to only 16% for the abstract-based approximations).

The presented methodology has been implemented in a proof of concept, as follows: the used e-reader updates important locations based on the current location of the reader and the locations derived from the content in the book. As seen in Figure 3, a map is shown upon request, together with the location of the user, and the locations of the relevant concepts in the current fragment the user is reading.

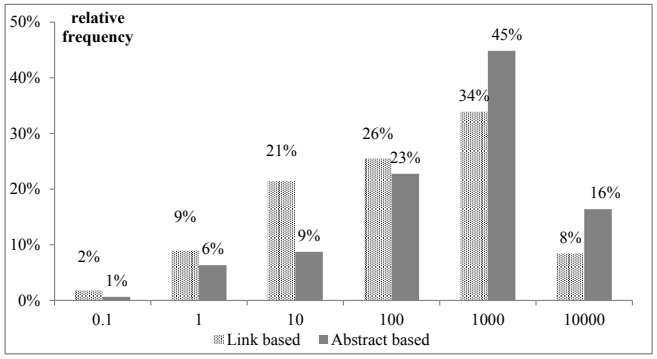

Figure 2: Histogram showing the relative amount of average errors (in kilometers), with a logarithmic bin interval.

These locations have a radius depending on the standard deviation of the approximation.

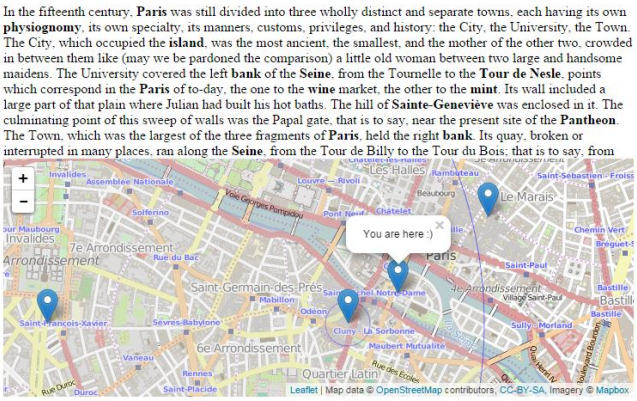

Figure 3: A screenshot of the proof of concept, showing relevant locations based on the actual location of the reader and the visible concepts in the publication

By only making use of the live DBpedia endpoint to calculate the approximations of the locations, we made a true Linked Data application – instead of making use of a cached local version. Web services such as OpenStreetMap do not link the saved locations with other resources on the Web, and GeoNames makes use of non-open data (i.e., their APIs are restricted in use), thus, a custom methods to calculate coordinates based on a DBpedia resource was devised.

When executing the same algorithm with `http://dbpedia.org/Wolgang_Amadeus_Mozart`, we get an approximation of N 48.1000 E 15.5250, with a radius of about $161.14km$. This

is located near Mozart's birth and death place, and is as such expected behavior. Furthermore, the radius gives the user a hint of the spread of that location. This results in a novel way of connecting digital publications with their readers: based on location. When someone is reading, e.g., The Hunchback of the Notre Dame by Victor Hugo, the results of this research could be used to hint the user to important locations close by, whether this is because an important scene in the book took place on that location, or because the author was born there. This results in a reconnection of the digital published works with the rest of the Web.

7. CONCLUSIONS AND FUTURE WORK

By processing digital publications using NER to detect named entities, and by saving these detected entities inside the publication using EPUB 3 and RDFa, we get an enriched book. By connecting the detected concepts with their location[18], and by connecting this location with the user, we get a novel location-based recommendation system, and we can reconnect digital publications with the rest of the Web.

The locations of concepts that do not have a location can be approximated using the links they have with other concepts that do have a location, or by using the detected entities in the abstract of the DBpedia resource. The evaluation shows that the link-based approach is generally more precise, but has a smaller recall. At the moment, a hybrid approach is recommended where the link-based approximation is calculated first, and the abstract-based approximation is used as a fallback. However, other hybrid solutions can be devised (e.g., combining the locations of both the linked concepts and the detected concepts in the abstract to have a true hybrid approximation, or, taking into account the correlation between standard deviation and error rate to pick the approximation with the smallest deviation), and should be evaluated. The results of the approximations could also be improved by omitting the outliers when calculating the average, to anticipate the approximation errors when a linked concept has a very different location than the root location.

Future work will focus on using this methodology to create more serendipitous discovery of relevant publications based on their location. Using the results of this work, the spatial center of gravity of a digital publication could be determined, and could be used to query relevant publications based on the location of the user. This will involve setting up an index of interlinked publications, together with their mentioned concepts and their (approximated) locations.

Acknowledgements.

The research activities described in this paper were funded by Ghent University, iMinds, the IWT Flanders, the FWO-Flanders, and the European Union, in the context of the project "Uitgeverij van de Toekomst" (*Publisher of the Future*).

8. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, and Z. Ives. DBpedia: A nucleus for a Web of Open Data. In *6th International Semantic Web Conference*, pages 11–15, Busan, Korea, 2007. Springer.

[2] S. S. Banerjee and R. R. Dholakia. Mobile advertising: does location based advertising work? *International Journal of Mobile Marketing*, 3(2):1–23, December 2008.

[3] G. Conboy, M. Garrish, M. Gylling, W. McCoy, M. Makoto, and D. Weck. EPUB 3 Overview. Technical report, IDPF, June 2014. Accessed January 22nd, 2015.

[4] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, 2013.

[5] I. Herman, B. Adida, M. Sporny, and M. Birbeck. RDFa 1.1 primer - second edition. Technical report, W3C, August 2013. Accessed January 22nd, 2015.

[6] I. Hickson. HTML Microdata. Technical report, W3C, October 2013. Accessed January 22nd, 2015.

[7] G. Klyne, J. J. Carroll, and B. McBride. RDF 1.1 concepts and abstract syntax. Technical report, W3C, February 2014. Accessed January 22nd, 2015.

[8] O. V. Laere, S. Schockaert, V. Tanasescu, B. Dhoedt, and C. B. Jones. Georeferencing wikipedia documents using data from social media sources. *ACM Trans. Inf. Syst.*, 32(3):12:1–12:32, July 2014.

[9] H. McGuire. A publisher's job is to provide a good api for books: you can start with your index. *The Indexer*, 31(1):36–38, March 2013.

[10] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[11] G. Rizzo and R. Troncy. NERD: Evaluating Named Entity Recognition tools in the Web of Data. In *Workshop on Web Scale Knowledge Extraction, ISWC2011*, Bonn, Germany, October 2011.

[12] D. A. Smith and G. Crane. Disambiguating geographic names in a historical digital library. In *Research and Advanced Technology for Digital Libraries*, pages 127–136. Springer, 2001.

[13] C. Stadler, J. Lehmann, K. Höffner, and S. Auer. Linkedgeodata: A core for a web of spatial open data. *Semantic Web Journal*, 3(4):333–354, 2012.

[14] R. Usbeck, A.-C. Ngonga Ngomo, S. Auer, D. Gerber, and A. Both. AGDISTIS - graph-based disambiguation of named entities using linked data. In *International Semantic Web Conference*. Springer, 2014.

[15] R. Verborgh, O. Hartig, B. De Meester, G. Haesendonck, L. De Vocht, M. Vander Sande, R. Cyganiak, P. Colpaert, E. Mannens, and R. Van de Walle. Querying datasets on the web with high availability. In *The Semantic Web–ISWC 2014*, pages 180–196. Springer, 2014.

[16] C. Wang, X. Xie, L. Wang, Y. Lu, and W.-Y. Ma. Detecting geographic locations from web resources. In *Proceedings of the 2005 workshop on Geographic information retrieval*, pages 17–24. ACM, 2005.

[18] An example installment is available at `http://uvdt.test.iminds.be:8990/`