

ECODIGIT

Ecosistema Digitale per la Fruizione e la Valorizzazione
dei Beni e delle Attività Culturali della Regione Lazio

D4.2 Strumenti per l'elaborazione semantica dei contenuti

Acronimo Progetto:

Titolo Progetto:

EcoDigit

**Ecosistema digitale per la fruizione
e la valorizzazione dei beni e delle
attività culturali della regione Lazio**

D4.2

Work Package:	WP4 T4.2	
Data di Sottomissione:	2 Ottobre 2019	
Inizio Progetto:	2 Ottobre 2018	
Durata Progetto:	15 Mesi	
Reponsabile Deliverable:	Valentina (ISTC-CNR)	Presutti
Versione:	1.0	
Stato:	Finale	
Autore:	Luigi Asprino	ISTC-CNR
	Ludovica Marinucci	ISTC-CNR
	Andrea Giovanni Nuzzolese	ISTC-CNR
	Valentina Presutti	ISTC-CNR
	Alessandro Russo	ISTC-CNR
Altri contribuenti al lavoro riportato nel deliverable:	Marco Canciani	(RM3)
	Marialuisa Mongelli	(ENEA)
	Luisa Carbone	(TUSCIA)
	Massimo Mecella	(RM1)
	Maria Prezioso	(RM2)
Reviewer:	Reviewer	Affiliazione

Per citare questo documento si prega di utilizzare il seguente record bibliografico

Luigi Asprino, Ludovica Marinucci, Andrea Giovanni Nuzzolese, Valentina Presutti e Alessandro Russo. *D4.2 Strumenti per l'elaborazione semantica dei contenuti*. Deliverable Progetto EcoDigit. 2019

Revisioni

Versione	Data	Modificata da	Commento
0.01	15/09/2020	Luigi Asprino	Prima Versione
0.02	25/09/2020	Luigi Asprino e Andrea Giovanni Nuzzolese	Aggiunta Bozza Sezione 1
0.02	30/09/2020	Alessandro Russo e Ludovica Marinucci	Aggiunta Bozza Sezione 2
0.03	15/10/2020	Valentina Presutti e Andrea Giovanni Nuzzolese	Revisione
0.04	30/10/2020	Luigi Asprino	Finalizzazione

Executive Summary

Le cinque università statali del Lazio in rete con CNR, ENEA e INFN si candidano a costituire il Centro di Eccellenza del Distretto Tecnologico per i beni e le attività Culturali (DTC) del Lazio. La mission del Centro di Eccellenza è costituire un centro di aggregazione ed integrazione di competenze nel settore delle tecnologie per i beni e le attività culturali. In questo contesto, il progetto **EcoDigit-Ecosistema digitale per la fruizione e la valorizzazione dei beni e delle attività culturali del Lazio** ha l'obiettivo di arricchire il sistema Anagrafe delle Competenze del DTC con una piattaforma middleware che faciliti l'integrazione di nuove sorgenti di dati e consenta la pubblicazione e il riuso di servizi per la valorizzazione e la fruizione del patrimonio culturale del Lazio.

L'obiettivo primario del Work Package 4 è identificare e integrare in ECODIGIT un insieme di applicazioni avanzate che rappresentino le punte di eccellenza nell'ambito della ricerca e delle competenze nel settore dei beni culturali.

Questo documento ha l'obiettivo di descrivere i servizi avanzati sviluppati nel contesto del Task T4.2 per l'arricchimento semantico dei contenuti testuali. In particolare verranno forniti i dettagli implementativi dei servizi per l'elaborazione automatica del testo quali: estrazione di entità, estrazione di grafi di conoscenza, collegamento automatico di entità provenienti da sorgenti differenti.

Indice

1	Introduzione	6
1.1	Obiettivi Work Package	6
1.2	Obiettivo del deliverable	6
1.3	Relazione con le altre attività del progetto	6
1.4	Outline documento	7
2	Servizi Avanzati di Entity Linking	8
2.1	Dettagli Implementativi	8
2.2	Entity Linking su Wikipedia	8
2.2.1	Dettagli implementativi	9
2.2.2	Esempi di Uso	9
2.3	Entity Linking su GeoNames	11
2.3.1	Dettagli implementativi	12
2.3.2	Esempi di Uso	12
3	Servizi Avanzati di Relation Extraction	15

Elenco delle tabelle

- 1 Esempio di richiesta al servizio di Entity Linking. I parametri *lang* e *sensorInventory* sono passati come URL parameter, mentre *text* nel body della richiesta. 10
- 2 Esempio di richiesta al servizio di Entity Linking. Il parametro *lang* è passato come URL parameter, mentre *text* nel body della richiesta. 12

1 Introduzione

1.1 Obiettivi Work Package

L'obiettivo primario del Work Package 4 è identificare e integrare in ECODIGIT un insieme di applicazioni avanzate che rappresentino le punte di eccellenza nell'ambito della ricerca e delle competenze nel settore dei beni culturali. L'approccio modulare è scelto per la sua facilità di fruizione e potenzialità di consultazione differenziata da parte di una ampia platea di destinatari, dal semplice utente all'amministrazione pubblica fino gli educatori e alle imprese del settore dei beni culturali, ed è supportato dall'architettura definita in WP2. L'obiettivo finale del WP è quello di sviluppare di un prototipo di fruizione dei Beni Culturali. In particolare si sta progettando un'interfaccia web nella quale è possibile accedere a contenuti semantici e ambienti virtuali, in particolar modo sviluppati nel dominio della formazione.

1.2 Obiettivo del deliverable

Questo documento ha l'obiettivo di descrivere i servizi avanzati sviluppati per l'arricchimento semantico dei contenuti testuali. In particolare verranno forniti i dettagli implementativi dei servizi per l'elaborazione automatica del testo quali: estrazione di entità, estrazione di grafi di conoscenza, collegamento automatico di entità provenienti da sorgenti differenti.

1.3 Relazione con le altre attività del progetto

Influenza degli altri task del progetto sul lavoro descritto in questo documento. Il lavoro descritto in questo documento si è avvalso dei risultati dei WP2 e WP3. In particolare, i moduli software descritti in questo documento applicano il paradigma di sviluppo software orientato a servizi definito nel WP2 (vedi task T2.1 [1]) e riusano componenti descritti nel documento architetturale [7]. Inoltre, lo sviluppo e la sperimentazione dei componenti descritti in questo documento si sono serviti dei dati raccolti a seguito dell'attività di censimento [6], trasformati con gli strumenti descritti nel Deliverable D3.3 [3] (processo documentato nel Deliverable D3.4 [4]), modellati secondo il formato definito nel Deliverable D3.2 [2]. I dati estratti/generati dagli strumenti descritti in questo documento sono rispettano il modello di ingresso definito nel Deliverable D3.2 [2]. Infine, il lavoro svolto nel task T4.2 ha sfruttato lo studio dello stato dell'arte presentato nel Deliverable D4.1 [5].

Influenza del lavoro descritto in questo documento sugli altri task del progetto. I dati estratti/generati dagli strumenti descritti in questo documento verranno usati nel prototipo di servizio avanzato nel dominio della formazione (vedi Task T4.4).

1.4 Outline documento

Nella Sezione 2 verranno presentati i servizi avanzati di Entity Linking, mentre nella Sezione 3 verranno presentati i servizi avanzati di Relation Extraction.

2 Servizi Avanzati di Entity Linking

Con il termine Entity Linking si intende il task di assegnare un'identità univoca a entità (come persone o città) menzionate in un testo. Le entità riconosciute nel testo sono quelle presenti in una base di conoscenza (o repository) di riferimento. Ad esempio, data la frase "La temperatura di oggi a Roma è 18 gradi", gli strumenti di Entity Linking sono in grado di associare alla parola "Roma" nel testo l'entità che rappresenta la città di Roma in Wikipedia¹. Nel contesto di EcoDigit sono stati sviluppati ed integrati nel middleware due sistemi di Entity Linking basati su servizi disponibili allo stato dell'arte. Il primo, descritto nella sezione 2.2, permette di individuare nel testo in input le menzioni di entità presenti in Wikipedia. Il secondo, descritto nella sezione 2.3, permette di individuare le città menzionate presenti in GeoNames.

2.1 Dettagli Implementativi

Seguendo i requisiti tecnologici e architetturali definiti nel WP2 (vedi i Deliverable D2.1 [1] e D2.2 [7]) i servizi di Entity Linking sono stati sviluppati come servizi REST, distribuiti all'interno di un docker container. Il codice relativo a servizi di Entity Linking è disponibile come progetto open source al seguente repository GitHub².

I servizi di Entity Linking possono essere installati usando Docker³. E' stato definito un Dockerfile parametrico che si occupa di installare tutte le dipendenze necessarie per l'esecuzione del servizio. Il Dockerfile accetta tre parametri: la porta su cui il server di entity linking si metterà in ascolto (i.e. `SERVER_PORT`), la chiave per accedere al servizio TagMe e lo username da usare per accedere alle API di GeoNames. Una volta costruito il container con il comando `docker build`, può essere eseguito con il comando `docker run`. A questo punto il server si metterà in ascolto sulla porta `SERVER_PORT` e risponderà alle richieste POST sui path definiti per i servizi di Entity Linking.

Infine, considerando che i servizi di Entity Linking si appoggiano a servizi esterni, al fine di limitare il numero di chiamate verso tali servizi è stato implementato un meccanismo di caching che evita di ripetere più volte richieste uguali.

2.2 Entity Linking su Wikipedia

Wikipedia⁴ è un'enciclopedia online a contenuto libero, collaborativa e multilingue contenente più di 45 milioni di voci in 280 lingue. Associare le entità menzionate in un testo con le entità presenti in Wikipedia permette di arricchire il testo con informazioni ipertestuali e

¹<https://it.wikipedia.org/wiki/Roma>

²<https://github.com/ecodigit/entitylinking>

³<https://www.docker.com/>

⁴<https://it.wikipedia.org/>

quindi di offrire all'utente una migliore fruizione dei testi permettendo ad esempio di avere un collegamento diretto alle informazioni presenti in Wikipedia. Il servizio avanzato di Entity Linking sviluppato nel contesto del progetto EcoDigit è basato su TagMe⁵ [8], servizio di Entity Linking disponibile allo stato dell'arte. Il lavoro svolto nel task T4.2 è stato dedicato all'integrazione di questo servizio all'interno del middleware EcoDigit.

2.2.1 Dettagli implementativi

Una volta eseguito il container i servizi di Entity Linking su Wikipedia saranno in ascolto sulla porta `SERVER_PORT` e risponderanno alle richieste POST sui seguenti path:

1. `/wikipedia/extractEntities` dato un testo `t`, questo servizio restituisce la lista di tutte le URI delle entità menzionate in `t`.
2. `/wikipedia/extractMentions` dato un testo `t`, questo servizio restituisce la lista di tutte le URI delle entità menzionate in `t`, indicando anche il punto del testo in cui una certa entità è menzionata.

Entrambi i servizi prendono in input obbligatoriamente un testo passato (in un oggetto JSON) come body della richiesta post come segue:

```
{ "text": "Miles Davis è un musicista Jazz." }
```

Inoltre entrambi i servizi accettano due parametri opzionali:

1. `lang` che indica la lingua del testo passato nel body della richiesta. Valori accettati: `IT` (valore di default), per indicare che il testo passato è in lingua italiana; `EN`, per indicare che il testo passato è in lingua inglese.
2. `senseInventory` che indica il repository da cui prendere le entità da riconoscere nel testo: `Wikipedia` (valore di default), per indicare che si vuole ottenere le URI delle entità presenti su Wikipedia; `DBPedia`, per indicare che si vuole ottenere le URI delle entità di DBPedia⁶.

Entrambi i parametri possono essere passati "Query Parameter" della URL.

2.2.2 Esempi di Uso

Si supponga che venga inviata al sul path `/wikipedia/extractEntities` la richiesta descritta in Tabella 1. Il risultato ottenuto è il seguente.

⁵<https://tagme.d4science.org/tagme/>

⁶<https://wiki.dbpedia.org/>

lang	it
senseInventory	Wikipedia
text	Miles Davis è un musicista Jazz.

Tabella 1: Esempio di richiesta al servizio di Entity Linking. I parametri *lang* e *senseInventory* sono passati come URL parameter, mentre *text* nel body della richiesta.

```
{
  "entities": [
    "https://it.wikipedia.org/wiki/Miles_Davis",
    "https://it.wikipedia.org/wiki/Musicista",
    "https://it.wikipedia.org/wiki/Jazz"
  ]
}
```

Si supponga che venga inviata al sul path `/wikipedia/extractMentions` la richiesta descritta in Tabella 1. Il risultato ottenuto è il seguente:

```
{
  "text": "Miles Davis è un musicista Jazz.",
  "annotations": [
    {
      "begin": 0,
      "end": 11,
      "mention": "Miles Davis",
      "mentionedEntities": [
        {
          "score": 1.0,
          "entity": "https://it.wikipedia.org/wiki/Miles_Davis"
        }
      ]
    },
    {
      "begin": 17,
      "end": 26,
      "mention": "musicista",
      "mentionedEntities": [
        {
          "score": 0.07009346038103104,
          "entity": "https://it.wikipedia.org/wiki/Musicista"
        }
      ]
    }
  ]
}
```

```

    "begin": 27,
    "end": 31,
    "mention": "Jazz",
    "mentionedEntities": [
      {
        "score": 0.48969632387161255,
        "entity": "https://it.wikipedia.org/wiki/Jazz"
      }
    ]
  }
]
}

```

2.3 Entity Linking su GeoNames

I dati territoriali costituiscono un elemento fondamentale per la valorizzazione delle peculiarità di un territorio. Ad esempio, conoscere la collocazione geografica dei beni culturali offre la possibilità di definire politiche territoriali mirate per le peculiarità del territorio. A tal fine si è ritenuto necessario provvedere il middleware di un servizio di Entity Linking capace di riconoscere menzioni di entità territoriali all'interno di un testo, fornendo contestualmente la geolocalizzazione delle entità riconosciute. Come repository di entità territoriali è stato usato GeoNames. GeoNames è un database di dati geografici accessibile attraverso vari servizi web. Tra i vari servizi messi a disposizione da GeoNames verrà usato l'API di search che data una stringa testuale restituisce tutte le entità territoriali con nome uguale alla stringa passata in input⁷. Tuttavia questo servizio non è in grado di estrarre menzioni da un testo qualsiasi. Ad esempio è capace di associare alla stringa "Roma" la URI `gn:3169071`⁸, ma non è capace di associare niente alla parola "Roma" menzionata nella frase "Roma è la capitale d'Italia".

Per ottenere il comportamento desiderato il servizio di Entity Linking sviluppato effettua un pre-processamento della stringa passata in input al fine di identificare, attraverso un passo di Named Entity Recognition (NER) (vedi Deliverable D4.1 [5]), i token annotati come "Location" nel testo passato come input. Ad esempio nella frase "Roma è la capitale d'Italia" vengono riconosciute come Location le parole "Roma" e "Italia". Per effettuare tale preprocessamento, il servizio sviluppato usa Stanford CoreNLP⁹ [9] per la lingua inglese e TINT¹⁰ [12] per la lingua italiana.

⁷<http://www.geonames.org/export/geonames-search.html>

⁸Il prefisso gn è associato al valore <https://sws.geonames.org/>.

⁹<https://stanfordnlp.github.io/CoreNLP/>

¹⁰<http://tint.fbk.eu/>

lang	it
text	Roma è la capitale d'Italia.

Tabella 2: Esempio di richiesta al servizio di Entity Linking. Il parametro *lang* è passato come URL parameter, mentre *text* nel body della richiesta.

2.3.1 Dettagli implementativi

Una volta eseguito il container, i servizi di Entity Linking su GeoNames saranno in ascolto sulla porta `SERVER_PORT` e risponderanno alle richieste POST sui seguenti path:

1. `/geonames/extractEntities` dato un testo *t*, questo servizio restituisce la lista di tutte le URI delle entità territoriali menzionate in *t* con la loro geolocalizzazione.
2. `/geonames/extractMentions` dato un testo *t*, questo servizio restituisce la lista di tutte le URI delle entità territoriali menzionate in *t* con la loro geolocalizzazione, indicando anche il punto della frase in cui una certa entità è menzionata.

Entrambi i servizi prendono in input obbligatoriamente un testo passato nel body della richiesta POST (in formato JSON) come segue:

```
{ "text": "Roma è la capitale d'Italia." }
```

Inoltre entrambi i servizi accettano un parametro opzionale (i.e. *lang*) che può essere passato come query parameter nella URL. Il parametro *lang* indica la lingua del testo passato nel body della richiesta. Valori accettati: IT (valore di default), per indicare che il testo passato è in lingua italiana; EN, per indicare che il testo passato è in lingua inglese.

2.3.2 Esempi di Uso

Si supponga che venga inviata al sul path `/geonames/extractEntities` la richiesta descritta in Tabella 2. Il risultato ottenuto è il seguente:

```
{
  "entities": [
    {
      "geoLocated": true,
      "latitude": "41.90036",
      "URI": "https://sws.geonames.org/3169071/",
      "longitude": "12.49575"
    },
    {
      "geoLocated": true,
```

```

        "latitude": "42.83333",
        "URI": "https://sws.geonames.org/3175395/",
        "longitude": "12.83333"
    },
    {
        "geoLocated": true,
        "latitude": "41.96667",
        "URI": "https://sws.geonames.org/3169069/",
        "longitude": "12.66667"
    }
],
"text": "Roma è la capitale d'Italia"
}

```

Dove:

1. <https://sws.geonames.org/3169071/> indica il comune di Roma.
2. <https://sws.geonames.org/3169069/> indica la città metropolitana di Roma Capitale.
3. <https://sws.geonames.org/3175395/> indica l'Italia.

Si supponga che venga inviata al sul path `/geonames/extractMentions` la richiesta descritta in Tabella 1. Il risultato ottenuto è il seguente:

```

{
  "text": "Roma è la capitale d'Italia",
  "annotations": [
    {
      "begin": 0,
      "end": 4,
      "mention": "Roma",
      "mentionedEntities": [
        {
          "score": 0.5,
          "entity": {
            "latitude": "41.90036",
            "longitude": "12.49575",
            "uri": "https://sws.geonames.org/3169071/",
            "geoLocated": true
          }
        }
      ],
    },
    {
      "score": 0.5,

```

```

        "entity": {
            "latitude": "41.96667",
            "longitude": "12.66667",
            "uri": "https://sws.geonames.org/3169069/",
            "geoLocated": true
        }
    }
]
},
{
    "begin": 21,
    "end": 27,
    "mention": "Italia",
    "mentionedEntities": [
        {
            "score": 1.0,
            "entity": {
                "latitude": "42.83333",
                "longitude": "12.83333",
                "uri": "https://sws.geonames.org/3175395/",
                "geoLocated": true
            }
        }
    ]
}
]
}

```

3 Servizi Avanzati di Relation Extraction

L'idea alla base del middleware sviluppato nel progetto EcoDigit è fornire degli strumenti che permettano di integrare sorgenti eterogenee. Tuttavia, come è facile immaginare, queste sorgenti potrebbero contenere contenuti tra loro correlati o riferirsi alle medesime entità. Ad esempio, una banca dati potrebbe contenere le informazioni riguardanti un certo oggetto del patrimonio culturale (e.g. “L'anfiteatro Flavio”), mentre una seconda potrebbe contenere delle pubblicazioni scientifiche che parlano dello stesso oggetto. Ovviamente, sarebbe molto interessante dal punto di vista dell'utente, presentare queste informazioni in maniera integrata, cioè collegando le entità presenti nei delle diverse banche dati.

A seconda delle varie sfumature, questo task viene chiamato in letteratura in modi diversi: *entity deduplication o resolution* (e.g. [14]) nel caso in cui si voglia trovare in una base di conoscenza eventuali entità duplicate; *relation extraction o knowledge graph completion* (e.g. [11, 10]) quando si vogliono creare delle relazioni tra le entità presenti in una base di conoscenza. Tra i principali svantaggi di questo tipo di strumenti troviamo (i) la necessità di fare apprendimento da grandi moli dati (quindi se non si lavora su una massa critica di dati, non si ottengono buoni risultati); (ii) la quasi incapacità di trattare contenuti testuali che potrebbero essere presenti all'interno della base di conoscenza.

Pur essendoci dei limiti a queste nuove tecnologie si è ritenuto necessario integrare all'interno del middleware un componente capaci di calcolare la similarità tra diverse entità della base di conoscenza. Questi strumenti, basati su strumenti disponibili allo stato dell'arte, possono aiutare sia nella deduplicazione delle entità (quando la similarità di due entità è molto alto si potrebbe ipotizzare che le entità sono uguali), che nel collegamento tra entità correlate.

Il primo servizio sviluppato si basa su l'approccio sviluppato da Ristoski et al. [13] chiamato RDF2VEC. Mentre il secondo usa una tecnica chiamata Latent Semantic Indexing¹¹ usando l'implementazione presente in gensim¹². Gli strumenti sviluppati sono in una fase prototipale e si rimanda ad una successiva fase di test e consolidamento dei componenti. Il codice sviluppato è disponibile come progetto open source al seguente repository GitHub¹³

¹¹https://en.wikipedia.org/wiki/Latent_semantic_analysis

¹²<https://radimrehurek.com/gensim/models/lsmodel.html>

¹³<https://github.com/ecodigit/lgu-commons-knowledgeengineering>

Riferimenti bibliografici

- [1] Luigi Asprino, Valentina Anita Carriero, Ludovica Marinucci, Andrea Giovanni Nuzzolese, Massimo Mecella, Valentina Presutti e Miguel Ceriani. *D2.1 Documento di analisi dei requisiti e Survey delle tecnologie*. Deliverable Progetto EcoDigit. 2019.
- [2] Luigi Asprino, Ludovica Marinucci, Andrea Giovanni Nuzzolese e Valentina Presutti. *D3.2 Modello di ingresso*. Deliverable Progetto EcoDigit. 2019.
- [3] Luigi Asprino, Ludovica Marinucci, Andrea Giovanni Nuzzolese e Valentina Presutti. *D3.3 Studio sugli strumenti di supporto*. Deliverable Progetto EcoDigit. 2019.
- [4] Luigi Asprino, Ludovica Marinucci, Andrea Giovanni Nuzzolese, Valentina Presutti, Massimo Mecella e Miguel Ceriani. *D3.4 Proof-of-Concept*. Deliverable Progetto EcoDigit. 2019.
- [5] Antonio Budano, Luigi Asprino e Mauro Saccone. *D4.1 Censimento dei tool esistenti*. Deliverable Progetto EcoDigit. 2019.
- [6] Miguel Ceriani e Massimo Mecella. *D3.1 Report sul Censimento*. Deliverable Progetto EcoDigit. 2019.
- [7] Miguel Ceriani, Massimo Mecella, Mauro Saccone, Marco Puccini, Luigi Asprino e Chiara Chiarelli. *D2.2 Descrizione del Middleware (documento architetturale)*. Deliverable Progetto EcoDigit. 2019.
- [8] Paolo Ferragina e Ugo Scaiella. “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*. ACM. 2010, pagine 1625–1628.
- [9] Jenny Rose Finkel, Trond Grenager e Christopher D. Manning. “Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling”. In: *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*. A cura di Kevin Knight, Hwee Tou Ng e Kemal Oflazer. The Association for Computer Linguistics, 2005, pagine 363–370. URL: <https://www.aclweb.org/anthology/P05-1045/>.
- [10] Guoliang Ji, Kang Liu, Shizhu He e Jun Zhao. “Knowledge graph completion with adaptive sparse transfer matrix”. In: *Thirtieth AAAI conference on artificial intelligence*. 2016.
- [11] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu e Xuan Zhu. “Learning entity and relation embeddings for knowledge graph completion”. In: *Twenty-ninth AAAI conference on artificial intelligence*. 2015.
- [12] A. Palmero Aprosio e G. Moretti. “Italy goes to Stanford: a collection of CoreNLP modules for Italian”. In: *ArXiv e-prints* (set. 2016). arXiv: 1609.06204 [cs.CL].
- [13] Petar Ristoski, Jessica Rosati, Tommaso Di Noia, Renato De Leone e Heiko Paulheim. “RDF2Vec: RDF graph embeddings and their applications”. In: *Semantic Web 10.4* (2019), pagine 721–752.

- [14] Linhong Zhu, Majid Ghasemi-Gol, Pedro Szekely, Aram Galstyan e Craig A Knoblock. “Unsupervised entity resolution on multi-type graphs”. In: *International semantic web conference*. Springer. 2016, pagine 649–667.