Proceedings Paper:
Gao, J. and Mazumdar, S. (2016) Exploiting linked open data to uncover entity types. In: Gandon, F., Cabrio, E., Stankovic, M. and Zimmermann, A., (eds.) Semantic Web Evaluation Challenges: Second SemWebEval Challenge at ESWC 2015, Portorož, Slovenia, May 31 - June 4, 2015, Revised Selected Papers. Second SemWebEval Challenge at ESWC 2015, 31 May - 04 Jun 2015, Portorož, Slovenia. Communications in Computer and Information Science, CCIS 548 . Springer International Publishing , pp. 51-62. ISBN 9783319255170

https://doi.org/10.1007/978-3-319-25518-7_5

Exploiting Linked Open Data to Uncover Entity Types

Jie Gao and Suvodeep Mazumdar

OAK Group, Department of Computer Science, University of Sheffield, United Kingdom
{j.gao,s.mazumdar}@sheffield.ac.uk

Abstract. Extracting structured information from text plays a crucial role in automatic knowledge acquisition and is at the core of any knowledge representation and reasoning system. Traditional methods rely on hand-crafted rules and are restricted by the performance of various linguistic pre-processing tools. More recent approaches rely on supervised learning of relations trained on labelled examples, which can be manually created or sometimes automatically generated (referred as distant supervision). We propose a supervised method for entity typing and alignment. We argue that a rich feature space can improve extraction accuracy and we propose to exploit Linked Open Data (LOD) for feature enrichment. Our approach is tested on task-2 of the Open Knowledge Extraction challenge, including automatic entity typing and alignment. Our approach demonstrate that by combining evidences derived from LOD (e.g. DBpedia) and conventional lexical resources (e.g. WordNet) (i) improves the accuracy of the supervised induction method and (ii) enables easy matching with the Dolce+DnS Ultra Lite ontology classes.

1 Introduction

A vast amount of knowledge is made available in the form of text; text is easily understandable by humans, but not by machines: applications can access knowledge if it is made available in a structured form. Information Extraction techniques serve the purpose of extracting facts from text and represent them in a structured form. FreeBase[1] and DBpedia[2] are famous examples of an effort to produce large scale world knowledge in a structured form. The structured facts are quite useful in tasks like question answering [20,8], facilitating both understanding the question and finding the answer. For example, in order to answer the question *"Which personification in Marvel Comics Universe was created by Bill Mantlo and Mike Mignola?"*, the knowledge of relations include (?x *created-by* "Bill Mantlo") (?x *created-by* "Mike Mignola") (?x *is-a* ?y) (?y *type-of* "personification"). A wider application of relation data can be seen in the Wikipedia infoboxes and more recently in Google Knowledge Graph initiative [19]. The relation data comes from large knowledge bases, which can be represented using different formalisms. Resource Description Framework (RDF) is the industry standard, which is designed to provide a common data model to represent structured information on the Web. Services like DBpedia draw on Wikipedia info-boxes to create such large databases

[1] https://www.freebase.com/
[2] http://wiki.dbpedia.org/About

[12], which now has 3 billion RDF triples, 580 million of which are extracted from English Wikipedia.

Open Information Extraction (Open IE) systems aim to extract information without being constrained by pre-specified vocabularies. State-of-the-art Open IE systems, e.g. ReVerb [7] and NELL [5], have witnessed remarkable success. Compared with schema-driven IE, Open IE can usually gain broader coverage thanks to a lightweight logical schema, though the lack of proper schema or unique identifiers cause a fair amount of ambiguity in the extracted facts and further hinder the data linking across multiple data sources.

This paper is in response to Open Knowledge Extraction (OKE) Challenge[3] in order to fill the gap between Open IE and existing centralised knowledge bases. We present a tool[4] for (i) identifying the type of the given entity (known a priori) in the given definition context; (ii) create a owl:Class statement for defining each of them as a new class in the target knowledge base, (iii) create a rdf:type statement between the given entity and the new created classes, and (iv) align the identified types with Dolce+DnS Ultra Lite (DUL) ontology classes[5], if a correct alignment is available, to a set of given types. Our approach consists of three main steps: (i) learning (in a supervised fashion) a model to recognize the word(s) in the sentence that express the type for the given entity (ii) predicting one or multiple types for all recognized (in previous step) surface forms expressing types; (iii) aligning all identified types to a given ontology. Each component will be explained in detail in Section 3. Evaluation results and conclusions are presented in Section 4 and 5 respectively.

2 Related Work

Named Entity Recognition. Named Entity Recognition (NER) is closely related to type extraction that aims to locate and classify atomic elements in text into predefined categories such as the names of persons or biological species, organizations, locations, etc. Three broad categories of machine learning paradigm in NER[16] include supervised, semi-supervised and unsupervised techniques. Feature engineering plays a crucial role in NER and has been well studied for many years. However, the difference is that NER systems do not label nominal (e.g., identify *"fictional villain"* as a type of *"Personification"*) or associate nominal phrases to entities.

Relation Extraction. Current methodologies in building a relation extractor generally fall into three categories: pattern-based [10], supervised machine learning, semi- and un-supervised approaches respectively. A number of popular methods has recently emerged in the third category include bootstrapping (i.e., using seeds)[1], distant supervision[15] and unsupervised learning[14] from the web. Different from most of relation extraction tasks that need the presence of two entities, our goal is to identify the hypernym (i.e.,*instance-Of*) relation between a given entity and noun phrases.

Ontology Matching. Our approach in type alignment is inspired from current practice and research in the field of ontology matching [17]. In this paper, we explored the

[3] https://github.com/anuzzolese/oke-challenge

[4] Source code can be found at https://github.com/jerrygaoLondon/oke-extractor.

[5] http://stlab.istc.cnr.it/stlab/WikipediaOntology/

combination of the context-based techniques by the use of formal resource (i.e., linked data) in semantic level and the content-based matching by the use of terminological techniques including string metrics for lexical similarity and WordNet for word relation matching, with respect to the schema level alignment for the matching between identified entity types and DUL ontology classes.

Interlinking Open Data. Emergence of Linked Data (LD) has raised increasing attention in the pressing needs for interlinking vast amounts of open data sources [2]. On the one hand, linked data can be leveraged as an external source of information for ontology matching, with respect to the challenge of "matching with background knowledge" [9]. On the other hand, interlinking methods derived from ontology matching can facilitate the achievement of the promise of Semantic Web: the Web of interlinked data. Motivated by both the LD based alignment method [11] and state-of-the-art interlinking methods (e.g., Silk[4], RDF-AI[18]), particular attention is paid in our approach to evaluate the role of LD in type extraction and alignment.

3 Methodology

Our approach can be represented as three main phases: (i) *training*, (ii) *prediction*, (iii) *type annotation and alignment* as illustrated in following architecture diagram (Fig. 1). The gold standard data contains *definition sentences*, i.e. each sentence expresses the type of a certain given entity[6]. We pre-process the gold standard data, we perform feature extraction and feature enrichment and we learn a classifier to recognize the portion(s) of the sentence expressing the entity type (we learn hyperonym patterns). All type candidates are fed to the type annotator which annotates each surface form as a new *owl:Class* with generated URIs in the format of NIF 2.0[7]. The well-formed new owl classes are then associated (by *rdf:type*) with the target entity in the sentence. In the final phase, the type alignment component performs semantic integration based on domain ontology and DUL ontology by combining linked data discovering (LDD), terminological similarity computation (TSS) and semantic similarity computation (SSC). Aligned DUL classes will be associated with identified type by *rdfs:subClassOf*[8]. The rationale of each component implementation is discussed in detail below.

3.1 Type Induction

Type induction is treated as a classical machine learning task in this experiment. First, the training set is loaded, parsed and mapped from the underlying RDF model to object-oriented(OO) data models. Parsing and processing NIF2RDF data is implemented

[6] We use the training data encoded in NIF format provided by the challenge organisers in this experiment. The NLP Interchange Format (NIF) is an RDF/OWL-based format that aims to achieve interoperability between Natural Language Processing (NLP) tools, language resources and annotations.

[7] http://persistence.uni-leipzig.org/nlp2rdf/

[8] The *rdfs* stands for the namespace of RDF Schema (`http://www.w3.org/2000/01/rdf-schema#`)

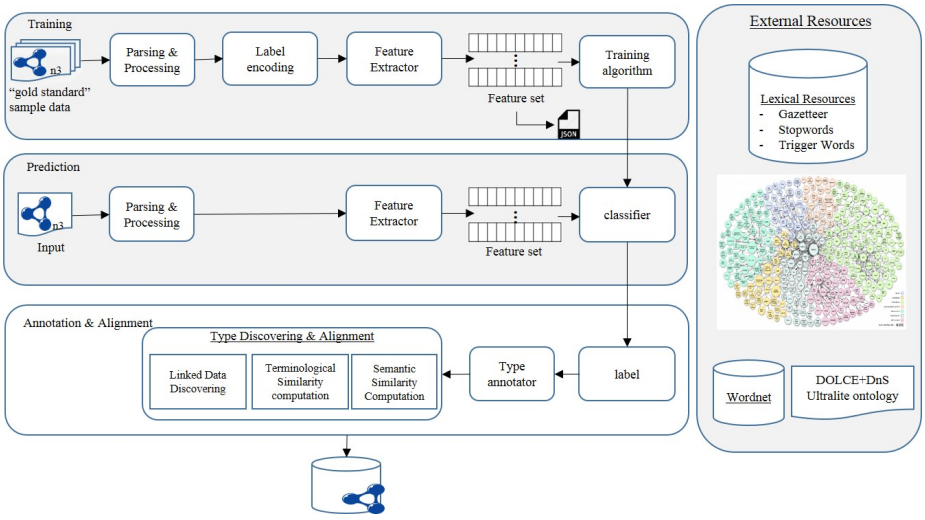

Fig. 1: Architecture of Type Induction and Alignment

on top of a general RDF library written in python (RDFLib)[9], which facilitates the parsing and serialisation of linked data in various formats. We implement a simple solution for this task that maps RDF model into an in-memory OO data model including *"TaskContext"*, *"ContextEntity"* and *"EntityClass"* respectively. Managing RDF data in an OO paradigm enables a quicker and more convenient data access model shared across multiple components.

Next, Context data (e.g., sentences, pre-labelled entities and types) are transformed and encoded in token-based data models $W = w_1, w_2, ..., w_n$, which treats each token (or word) as atomic unit (called hereafter data point). Each data point $w_i \in W$ represents a token (or word) with its feature set, its class label and its unique identifier. Each data point from the sentence is considered as a learning instance which is labelled with corresponding class labels. Following the approach of [13], we adopt a two-class IO labelling scheme, where each data point is either in-type (labelled as "I") or out-of-type (labelled as "O").

Feature Extraction In the feature extraction phase we construct the feature set for each data point. We collect the following features:

1. Word-level features: For each data point which is not a stopword[10] we produce: "WORD_POS": word PoS category; "IS_TITLE": true if the word is a titlecased string, "ALL_CAPITAL":true if all cased characters in word are uppercase, "IS_WORD_ROOT_BE": true if the lemma of current word is 'be';"IS_PUNCT_COMMA": true if current

[9] RDFLib: https://pypi.python.org/pypi/rdflib

[10] The SMART stop-word list built by Chris Buckley and Gerard Salton, which can be obtained from goo.gl/rBQNbO

word is a comma punctuation; "WORD_WITH_DIGITS": true if current word contains digits;"LAST_2_LETTERS": last two characters of current word.

2. Named-entity: We include the feature "IS_ENTITY" to indicate whether current word is entity or not.
3. Gazetteer and trigger word features: Trigger words are a list of terms that might be useful for relation extraction. For example, trigger words like *"Mr"*, *"Miss"* and *"Dr"* for Person, *"city"* and *"street"* for location, *"Ltd"* and *"Co."* for Organisations, are obviously useful to recognise the instance-of relations. We also hand-picked a list of trigger words (e.g., *"name"*, *"form"*, *"class"*, *"category"*, *"variety"*, *"style"*, *"model"* and *"substance"*) that can indicate the type relations. WordNet can be employed to extract trigger words, e.g., look for synonyms. Gazetteer features can be a list of useful geo or geopolitical words e.g., country name list and other sub-entities such like person first name, person surname. We used the ANNIE Gazetteer[11] from GATE platform [12] in our experiment. A list of gazetteer based features used include "TYPE_INDICATOR": true if current word is matched with an item in type trigger words; "IS_STOPWORD":true if current word is stop word; "IS_ORGKEY":true if current word is matched with an item in organisation entity trigger words; "IS_LOCKEY":true if current word is matched with an item in location entity trigger words; "IS_COUNTRY": true if current word is country entity; "IS_COUNTRYADJ":true if current word is country adjective; "IS_PERSONNAME":true if current word is person name trigger words (e.g., firstname, surname); "IS_PERSONTITLE": true if current word is person title;"IS_JOBTITLE":true if current word is job title entity; "IS_FACKEY": true if current word is facility entity trigger words.
4. Neighborhood features: We include surrounding words and their corresponding features; this provides contextual evidence useful to discover hypernym pattern between identified entities and the target word expressing the type. Position information is encoded in the feature names and examples of such feature set are "PREV_2_WORD_WITH_DIGITS", "NEXT_1_WORD_IS_STOPWORD", "PREV_1_WORD_POS", "PREV_3_WORD_IS_COUNTRY" and so forth. In our experiment, features are extracted from a 8×3 sliding window.
5. semantic distance: The'SEMANTIC_DISTANCE" is a numerical value which quantifies the "similarity of meaning" between the target token t_1 , i.e. the word(s) potentially expressing the types, and the target entity t_2, i.e. the one for which the type is being expressed. The value is computed by looking at all possible types that we can gather for t_1 and t_2 from LOD (specifically DBpedia). Formally, the semantic distance is computed as:

$$sem\_dist(t_1, t_{2j}) = max[sim(S_n(t_1), S_n(t_{2j}))], t_{2j} \in rdf : type(E), n > 0 \quad (1)$$

t_1 is the target token and t_{2j} is the one of linked data types (rdf:type) associated with entity (E). As entity is disambiguated by Dbpedia URI in the dataset, we can acquire that disambiguated type information by SPARQL query. S_n is the synset of a word where several meanings of the word can be looked up. *sim*() is

[11] https://gate.ac.uk/sale/tao/splitch13.html
[12] https://gate.ac.uk/

the maximum semantic similarity determined by the function of the path distance between words in hierarchical structure in Wordnet. Our assumption is based on the fact that existing resources like WordNet and DBpedia are a rich and reliable source of hyponymy/hypernymy relationships between entities, which are assumed to be able to provide very informative and potentially strong indications about instance-of relation between entity and target token. Even though type information is usually multi-word terms, our intuition is to identify head noun in multi-word type surface form. This is based on the assumption that terminological heads usually carry key conceptual semantics[6]. We implemented the sem_dict() based on NLTK WordNet[13] library and python SPARQLWrapper[14] for rdf type and label query. The semantic similarity is computed by the WordNet *path_similarity* function which is based on the shortest path connecting the word senses in the is-a (hyernym/hypnonym) taxonomy.

Model selection We experimented with three state-the-art classifiers, including Naïve Bayes, Maximum Entropy Markov Model (MEMM) and Support Vector Machine Model provided in NLTK's classify package[15]. Based on the same feature set and 100 iterations, our experiment indicates that even if Naïve Bayes classifier and SVM is much fast in training, MEMM give us the optimum performance for our class induction task. Moreover, as a discriminative classifier, more features make MEMM model more accurate.

Type annotation In order to identify all possible type surface forms for a certain entity, we combined the approach of head noun extraction and the PoS based grammar matching for compound words combining the modifiers and a head noun. For the above example, the continuing tokens ‘‘*American lightweight boxer"* can be picked out with type tag "I" after processed with type classifier, while ‘‘*lightweight boxer"* and ‘‘*boxer"* are also good candidate entity types. A set of PoS patterns grammars (Table 1) are applied iteratively in our experiment. Note that + and * are regular expression cardinality operators. PoS-tagging was achieved with the NLTK standard treebank POS tagger[16].

<JJ|VBG|VBD>+ <NN|NNP|NNS>+
<NN|NNP|NNS>+ <JJ|VBD|VBG>* <NN|NNP|NNS>+

Table 1: A simplified version of PoS grammar patterns matching multiple type surface forms

[13] http://www.nltk.org/howto/wordnet.html
[14] SPARQLWrapper is a python based wrapper around a SPARQL service, access via http://rdflib.github.io/sparqlwrapper/
[15] http://www.nltk.org/howto/classify.html
[16] http://www.nltk.org/book/ch05.html

3.2 Type Alignment

The motivation of class alignment method in our experiment is to investigate how LOD datasets (typically DBpedia) can facilitate the alignment of heterogeneous type information. Our alignment method is based on the heuristics that the linked data resource is typed and linked by their dereferenceable URIs. For example (in Figure 2), to identify whether a football club is type of "dul:Agent"[17], we can ask this question based on LOD knowledge base (typically DBpedia in our case), which can be constructed in the following SPARQL query.

Fig. 2: Parts of Extracted Entity Class and DUL classes

```
ASK {
    ?instance dbpedia-owl:type ?entity.
    <http://dbpedia.org/resource/Football_Club> dbpedia-owl:wikiPageRedirects ?entity.
    ?instance a ?type.
    FILTER(?type = dul:Agent)
}
```

In the task of DUL ontology alignment, early experiments show that there are 9% (9 out of 99 entities) DBpedia entities in the gold standard dataset are classified with DUL classes. By using dereferenceable type URI with a more complex SPARQL query [18], we found that about 30% (60 out of total 201) types can be directly matched with DUL classes pre-classified in DBpedia. If counting all the multi-word types containing the matched head nouns, there are 117 types (58.2% of total) that can be aligned with DUL classes via DBpedia. A typical example as above, if "Club" is directly matched with "dul:Agent" via query, "Football Club" containing "Club" as the head noun can be further aligned with "dul:Agent".

Our alignment process can be divided into three steps: linked data discovery, terminological similarity computation and semantic similarity computation. Linked Data Discovery (LDD) is essentially the semantic query based on existing structural knowledge in DBpedia. We combine multiple classification schemes from DBpedia about the entity and extracted classes to determine best matched DUL classes. Entity based query

[17] The *dul* stands for the prefix for http://www.ontologydesignpatterns.org/ont/dul/DUL.owl.
[18] The complete SPARQL query can be found in the projects source code repository

is achieved by the DBpedia URI and the corresponding DUL classes about extracted entity types can be retrieved by automatically generated dereferencing URI following the practice in DBpedia [3]. Multi-word type terms that contain the matched head noun type in the same context will be aligned with the same DUL class. For many cases that no DUL classes can be found by LDD, we compute terminological similarity by Levenshtein distance normalised by the length of the longest sequence. The threshold is set to 0.9. The schema level matching is based on the lexicon expansion on both target class and DUL classes to be aligned. Target class is expanded by type labels extracted from both entity and dereferencable type from DBpedia. Meanwhile, DUL classes are expanded with keywords and synonyms. Table 2 illustrates the parts of DUL classes and keywords.

DUL Classes	**Keywords & synonyms**
dul:Activity	activity, task
d0:Characteristic	characteristic, feature
d0:CognitiveEntity	Attitudes, cognitive, ideologies, mind
dul:Goal	Goal, aim, achievement
d0:Location	Place, space
dul:Organism	Organism, animal, plant
dul:Personification	personification, fictional, imaginary
dul:Situation	situation, condition, circumstance, state

Table 2: Parts of DUL Classes and Keywords

In the final step of our method, for the classes that cannot be aligned by string similarity, we adopt the semantic similarity computation approach that relies on semantic taxonomy in WordNet to determine hypernym relationship between expanded target type and expanded DUL classes labels. For multi-word terms, we compute the similarity based on head noun. Where either or both of the words had more than one synset in WordNet, we compute all the combinations to find the synsets pair of maximal similarity. The similarity threshold (i.e.,path distance) is set to 0.1.

4 Evaluation

4.1 Type Induction

For the experiment of type induction task, the gold-standard corpus was used which contains 99 sentences of entity definition context. The gold-standard corpus is split into 70% for training and 30% for testing. The performance of entity type extraction is computed in by Precision (P), Recall(R), and F-measure (F1 score) (as follows).

$$P = \frac{\#TruePositive}{\#TruePositive + \#FalsePositive} \tag{2}$$

$$R = \frac{\#TruePositive}{\#TruePositive + \#FalseNegative} \tag{3}$$

$$F_1 = \frac{2PR}{P + R} \tag{4}$$

As shown in Table 3, the MEMM classifier trained with features not derived from LD source is used as baseline for performance comparison. By add the LD based feature "SEMANTIC_DISTANCE" to train MEMM model achieve overall 5.19 increase of F-score, with 1.02 and 6.38 increase in precision and recall.

	P(%)	R(%)	F1(%)
MEMM without LD features	84.23	47.10	60.27
MEMM with LD features	85.25	53.48	65.46

Table 3: Results of Evaluation of Class Induction Method

4.2 Type Alignment

Type alignment evaluation is implemented as follows.

$$P = \frac{\#correctIdentifiedAlignments}{\#identifiedAlignments} \tag{5}$$

$$R = \frac{\#correctIdentifiedAlignments}{\#goldStandardData} \tag{6}$$

$$F_1 = \frac{2PR}{P + R} \tag{7}$$

In (5) and (6), the "#correctIdentifiedAlignments" is computed by combining string matching and subsumption reasoning. Specifically, if automatically aligned DUL types cannot be matched with labelled data (i.e., gold standards), we check whether the DUL type is the subclass of the labelled DUL type or vice versa. In other words, if at least one gold standards alignment can be matched lexically or semantically, the result is recognised as correct. We compared three different alignment strategies and a combination of two or three of them in Table 4.

From the evaluation results, even if LDD has good coverage 63% (62 out of 99) for alignment suggestions, the performance of the LDD method has a low overall F-measure. TSC method achieved higher performance than LDD and SSC, which has further gained 2.45% improvement with optimal result by combining with SSC.

	P(%)	R(%)	F1(%)
Linked Data Discovering (LDD)	35.48	22.2	27.33
Terminological Similarity Computation (TSC)	75.44	43.43	55.13
Semantic Similarity Computation (SSC)	38.38	38.38	38.38
TSC + SSC	57.58	57.58	57.58
LDD+TSC+SSC	34.34	34.34	34.34

Table 4: Results of Evaluation of Type Alignment Method

4.3 Competition Result

The overall performance evaluated in official competition[19] is presented in Table 5

Annotator	Micro F1	Micro Precision	Micro Recall	Macro F1	Macro Precision	Macro Recall
CETUS	0.4735	0.4455	0.5203	0.4478	0.4182	0.5328
OAK@Sheffield	0.4416	0.5155	0.39	0.3939	0.3965	0.3981
FRED	0.3043	0.2893	0.3211	0.2746	0.2569	0.3173

Table 5: Official Competition Results of OKE Task 2

5 Conclusion

Linked Open Data opens up a promising opportunity for machine learning in terms of feature learning from large scale and ever-growing graph-based knowledge sources. In this paper, we present a hybrid approach for automatic entity typing and type alignment. We experimented three different strategies in type alignment. The evaluation result suggests that LOD can complement extremely rich semantic information compared with WordNet, particularly for complex multiword schema terms. Even though the type alignment directly suggested by LOD suffers low quality, the corresponding concept hierarchies from the multiple community-driven classification schemes can contribute very effective semantic evidences for facilitating alignment task with respect to the similarity and relatedness measurement.

Acknowledgments Part of this research has been sponsored by the EPSRC funded project LODIE: Linked Open Data for IE, EP/J019488/1

References

1. Agichtein, E., Gravano, L.: Snowball: Extracting relations from large plain-text collections. In: Proceedings of the Fifth ACM Conference on Digital Libraries. pp. 85–94. DL '00, ACM, New York, NY, USA (2000), http://doi.acm.org/10.1145/336597.336644

[19] https://github.com/anuzzolese/oke-challenge

2. Bizer, C., Heath, T., Ayers, D., Raimond, Y.: Interlinking Open Data on the Web. Media 79(1), 31–35 (2007), http://people.kmi.open.ac.uk/tom/papers/bizer-heath-eswc2007-interlinking-open-data.pdf
3. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia-a crystallization point for the web of data. Web Semantics: science, services and agents on the world wide web 7(3), 154–165 (2009)
4. Bizer, C., Volz, J., Kobilarov, G., Gaedke, M.: Silk - a link discovery framework for the web of data. In: 18th International World Wide Web Conference (April 2009), http://www2009.eprints.org/227/
5. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: In AAAI (2010)
6. Daille, B., Habert, B., Jacquemin, C., Royauté, J.: Empirical observation of term variations and principles for their description. Terminology 3(2), 197–257 (1996)
7. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1535–1545. EMNLP '11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011), http://dl.acm.org/citation.cfm?id=2145432.2145596
8. Fader, A., Zettlemoyer, L., Etzioni, O.: Open question answering over curated and extracted knowledge bases. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1156–1165. KDD '14, ACM, New York, NY, USA (2014), http://doi.acm.org/10.1145/2623330.2623677
9. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: Discovering missing background knowledge in ontology matching. In: Proceedings of the 2006 Conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29 – September 1, 2006, Riva Del Garda, Italy. pp. 382–386. IOS Press, Amsterdam, The Netherlands, The Netherlands (2006), http://dl.acm.org/citation.cfm?id=1567016.1567101
10. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora (1992)
11. Kachroudi, M., Moussa, E.B., Zghal, S., Ben, S.: Ldoa results for oaei 2011. Ontology Matching p. 148
12. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web Journal (2014)
13. Li, Y., Bontcheva, K., Cunningham, H.: Adapting svm for data sparseness and imbalance: a case study in information extraction. Natural Language Engineering 15(02), 241–271 (2009)
14. Min, B., Shi, S., Grishman, R., Lin, C.Y.: Ensemble semantics for large-scale unsupervised relation extraction. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 1027–1037. Association for Computational Linguistics (2012)
15. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2. pp. 1003–1011. ACL '09, Association for Computational Linguistics, Stroudsburg, PA, USA (2009), http://dl.acm.org/citation.cfm?id=1690219.1690287
16. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes 30(1), 3–26 (Jan 2007), http://dx.doi.org/10.1075/li.30.1.03nad
17. Otero-Cerdeira, L., Rodríguez-Martínez, F.J., Gómez-Rodríguez, A.: Ontology matching: A literature review. Expert Systems with Applications 42(2), 949 – 971 (2015), http://www.sciencedirect.com/science/article/pii/S0957417414005144

18. Scharffe, F., Liu, Y., Zhou, C.: Rdf-ai: an architecture for rdf datasets matching, fusion and interlink. In: Proc. IJCAI 2009 workshop on Identity, reference, and knowledge representation (IR-KR), Pasadena (CA US) (2009)
19. Singhal, A.: Introducing the knowledge graph: things, not strings. Official Google Blog (May 2012)
20. Yao, X., Van Durme, B.: Information extraction over structured data: Question answering with freebase. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 956–966. Association for Computational Linguistics (2014), http://aclweb.org/anthology/P14-1090