Extending the Coverage of DBpedia Properties using Distant Supervision over Wikipedia

Alessio Palmero Aprosio[1], Claudio Giuliano[2], and Alberto Lavelli[2]

[1] Università degli Studi di Milano, Via Comelico 39/41, 20135 Milano, Italy
alessio.palmero@unimi.it
[2] Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy
{giuliano,lavelli}@fbk.eu

Abstract. DBpedia is a Semantic Web project aiming to extract structured data from Wikipedia articles. Due to the increasing number of resources linked to it, DBpedia plays a central role in the Linked Open Data community. Currently, the information contained in DBpedia is mainly collected from Wikipedia infoboxes, a set of subject-attribute-value triples that represents a summary of the Wikipedia page. These infoboxes are manually compiled by the Wikipedia contributors, and in more than 50% of the Wikipedia articles the infobox is missing. In this article, we use the distant supervision paradigm to extract the missing information directly from the Wikipedia article, using a Relation Extraction tool trained on the information already present in DBpedia. We evaluate our system on a data set consisting of seven DBpedia properties, demonstrating the suitability of the approach in extending the DBpedia coverage.

1 Introduction

Wikipedia is one of the most popular web sites in the world and the most used encyclopedia. In addition, Wikipedia is steadily maintained by a community of thousands of active contributors, therefore its content represents a good approximation of what people need and wish to know. Finally, Wikipedia is totally free and it can be downloaded entirely thanks to periodic dumps made available by the Wikipedia community. For these reasons, in the last years several large-scale knowledge bases (KB) have been created exploiting Wikipedia. DBpedia [1], Yago [22] and FreeBase [3] are relevant examples of such resources.

In this work, we are particularly interested in DBpedia.[3] Created in 2006, DBpedia has grown in size and popularity, becoming one of the central interlinking hubs of the emerging Web of Data. The approach adopted to build DBpedia is the following. First, the DBpedia project develops and maintains an ontology, available for download in OWL format. Then, this ontology is populated using a rule-based semi-automatic approach that relies on Wikipedia *infoboxes*, a set of *subject-attribute-value* triples that represents a summary of some unifying aspect that the Wikipedia articles share. For example, biographical articles typically have a specific infobox (`Persondata` in the English Wikipedia) containing information such as *name*, *date of birth*, *nationality*,

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

activity, etc. Specifically, the DBpedia project releases an extraction framework used to extract the structured information contained in the infoboxes and to convert it into triples. Moreover, crowdsourcing is used to map infoboxes and infobox attributes to the classes and properties of the DBpedia ontology, respectively. As the number of required mappings is extremely large, the whole process follows an approach based on the frequency of infoboxes and infobox attributes. Most frequent items are mapped first. This guarantees a good coverage because infoboxes are distributed according the Zipf's law [23]. This mapping process is divided into two different steps. First, the infobox is mapped to the corresponding class in the DBpedia ontology; then, infobox attributes are mapped to the properties owned by that class. Using the extraction framework provided by the DBpedia community, pages containing such infobox are automatically assigned to the class, and for each page the infobox attributes are used to populate the corresponding properties. For example, the `Infobox_journal` infobox is mapped to the `AcademicJournal` DBpedia class, and its attributes `title`, `editor`, and `discipline` are then mapped, respectively, to the properties `foaf:name`, `editor`, and `academicDiscipline`. Finally, the resulting KB is made available as Linked Data,[4] and via DBpedia's main SPARQL endpoint.[5]

At the time of starting the experiments reported in this paper, the version of the English DBpedia available (3.8) covers around 1.7M entities, against almost 4M articles in Wikipedia. The main reason for this problem of coverage is the lack of infoboxes for some pages (and also the limited coverage of the mappings mentioned above). In fact, even if Wikipedia provides an infobox suitable for a page, it may happen that the users who write the article do not know how to specify it in the source code of the page, or simply they do not know that infoboxes (or that particular infobox) exist.

Recently, in 2013, a project called Airpedia [16] addressed the problem of extending the DBpedia coverage with respect to classes. The working hypothesis is that the article class (i.e. `Person`, `Place`, etc.) can be inferred by some features of the page, for example the list of categories and the latent semantic analysis of the text.

In this paper, we extend this approach to properties. There are projects aiming to extract properties from some structured parts of the page different from infoboxes. For example, Yago exploits categories [22]. However, such approach is feasible only for a small number of attributes (for example the Wikipedia page `Barack_Obama` is included in the `1961_births` category, from which it can be inferred that Obama's birth year is 1961). Therefore, to populate the whole DBpedia set of properties (over 1,500 in version 3.8), we need to find the relevant information right from the page article, using Natural Language Processing (NLP) tools. This is a relation extraction (RE) task, i.e. the identification in a text of relevant entities and relationships between them. For example, given the sentence "Barack Obama was born on August 4, 1961", we need to identify "Barack Obama" as a named entity of type person, the value "August 4, 1961" as a date, and the `birthDate` relation between the two objects.

Supervised machine learning techniques are widely used to approach the RE task, but the lack of manually annotated texts to use as training data often limits the applicability of such techniques. In 2009, a new paradigm, called *distant supervision* [14], has been

[4] `http://wiki.dbpedia.org/Downloads`
[5] `http://dbpedia.org/sparql`

proposed to deal with the limited availability of manually annotated data. The intuition behind distant supervision is that any sentence containing a pair of entities that participate in a known DBpedia property relation is likely to express such relation in some way. Using the example above, the assumption is that a sentence that includes both "Barack Obama" and "August 4, 1961" is expressing the `birthDate` relation. Since there are thousands of such pairs of entities in the DBpedia resource, we can extract very large numbers of (potentially noisy [19]) examples for each relation.

In this work, we first collect this set of sentences starting from DBpedia and extracting the relevant sentences from the corresponding Wikipedia articles. Then, we train a RE tool (jSRE [8], freely available on the web) using positive and negative examples extracted from such sentences. Finally, we apply the model on unseen text articles and extract the relations contained in the sentences.

We evaluate our system on seven DBpedia properties, using cross-validation over a small set of pages excluded from the training, demonstrating the suitability of the approach with high precision and recall.

The work reported in this paper is part of a wider effort devoted to the automatic expansion of DBpedia [16–18] and tackles the issue of extracting properties for those pages where the DBpedia paradigm cannot be applied (for example when the infobox is missing). Table 1 summarizes the different steps of such effort on DBpedia expansion.

Table 1. Various steps for automatic expansion of DBpedia.

	Automatic mapping for new languages	Instance-based classification
Classes	[17]	[16]
Properties	[18]	this paper

2 Related work

The main reference for our work is the DBpedia project [1]. Started in 2006, its goal is to build a large-scale knowledge base semi-automatically using Wikipedia. Differently, Yago [22], another similar project started in 2007, aims at extracting and mapping entities from Wikipedia using categories (for fine-grained classes) and WordNet (for upper-level classes). Yago uses particular categories to map properties (for example the Wikipedia page `Albert_Einstein` is included in the `American_physicists` category, from which it can be inferred that Albert Einstein's occupation is physicist), but this method can cover only a small number of relations (52 in Yago 2 [22]) and a big effort is needed to port it to other languages. Conversely, FreeBase [3] and WikiData [26] are collaborative knowledge bases manually compiled by their community members.

The distant supervision paradigm [14], presented in 2009, has been widely used for RE purposes [15, 11, 7]. The basic assumption that each sentence which mentions the entities involved in the relation is an expression of the relation itself produces the effect

that noisy examples can be present in the training. In [19] a new approach to distant supervision is proposed, dealing with the presence of noisy examples in the training. A survey on noise reduction methods for distant supervision is discussed in [20].

In addition, distant supervision has been recently tested on the web, to obtain rule sets large enough to cover the actual range of linguistic variation, thus tackling the long-tail problem of real-world applications [10], for sentiment analysis in social networks [9], fact checking [13], and question answering [5].

Finally, the problem of an automatic expansion of the DBpedia dataset is tackled from various perspectives. On the one hand, starting from version 3.7, DBpedia is available for different languages. As the corresponding versions of Wikipedia do not share the same infobox structure, the manual effort needed to manually build mappings needs to be multiplied by the number of versions of DBpedia. Some automatic approaches to this problem has been proposed in the last year [17, 18]. On the other side, DBpedia only considers Wikipedia pages that contain an infobox (and for which the mapping was provided). The Airpedia project [16] deals with this problem, using a machine learning approach to guess the DBpedia ontology class of a page using features extracted from the entire Wikipedia page.

3 Workflow

As introduced before, the work presented in this paper relies on the intuition that jointly exploiting interlinked structured and unstructured data sources can offer a great potential for both NLP and Semantic Web applications. In particular, we focus on the pair Wikipedia-DBpedia as corpus-KB and on distant supervision as paradigm.

Wikipedia[6] is a collaboratively constructed online encyclopedia: its content is free to reuse, and can be edited and improved by anyone following the rules of the edition process. Wikipedia articles can also contain structured elements like the *infobox*, providing factual data in the form of values associated to fields, that can be easily extracted.

DBpedia is a database derived from Wikipedia. Since Wikipedia pages contain a lot of potentially useful data, the Semantic Web community started an effort to extract data from Wikipedia infoboxes and then to publish them in a structured format (following the open standards of the Semantic Web) to make them machine-readable. DBpedia releases its dataset in RDF format, a conceptual model expressed in the form of subject-predicate-object. Example of an instance of the dataset (triple) is

$$\langle \text{Barack Obama} \rangle \, \langle \text{is\_born\_in} \rangle \, \langle \text{Honolulu} \rangle \tag{1}$$

where "Barack Obama" is the subject, "is_born_in" is the predicate, and "Honolulu" is the object. The set of possible values of subject and object are called *domain* and *range*, respectively. In our experiments, we only consider triples involved in relations between entities, expressing DBpedia properties, therefore the predicate represents the relation/property, and the subject-object pair refers to the entities involved in that relation. In particular, in DBpedia the subject of such a triple is always related to a Wikipedia page.

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

The distant supervision paradigm is based on the assumption that there is a high probability that the structured information present in the *infobox* is also expressed using natural language sentences in the same Wikipedia page. Given the triple (1), we expect that there is a sentence in the text article expressing the same relation with the same entity mentions, like

> Obama was born on August 4, 1961 at Kapiolani Maternity & Gynecological Hospital in Honolulu, Hawaii, and is the first President to have been born in Hawaii.

Summarizing, for each DBpedia property we apply the following procedure:

1. all the triples expressing the relation are considered;
2. for each triple, the corresponding Wikipedia article is analyzed using Stanford CoreNLP (Section 4);
3. the sentences containing both the subject and the object of the triple are extracted and collected as positive examples (Section 5.1);
4. a set of negative examples is collected, too (Section 5.2);
5. a RE tool is trained using the dataset built according to the procedure outlined above (Section 5.3);
6. the trained model is then applied to extract the desired relation from article pages where the infobox is missing or where the infobox does not contain such relation.

The main part of the procedure is the RE task. Formally, given a sentence S which consists of a sequence of words, we need to find a relation R that involves two sequences of words E_1 and E_2, respectively the subject and the object of the relation. In our experiments, we use jSRE,[7] a state-of-the-art open source RE tool, made freely available on the web.

To assess to performance of the approach, we test the accuracy of the entire workflow on a dataset consisting of seven DBpedia properties (Section 6).

4 Pre-processing

The preliminary phase of our approach consists in collecting Wikipedia pages where a particular relation is (likely to be) expressed. The list of such pages can be easily extracted from DBpedia.

Given the Wikipedia page, the plain text article is obtained by removing tables, images and all the wiki markup. We use JWPL[8] for such purpose.

The cleaned up text is then analyzed using Stanford CoreNLP[9] with the following processors: tokenization, sentence splitting, part-of-speech tagging, lemmatization and Named Entity Recognition (NER). In particular, the NER module of Stanford CoreNLP annotates persons, organizations, locations, numbers and dates. In addition, we use the

[7] `http://hlt.fbk.eu/en/technology/jSRE`
[8] `https://code.google.com/p/jwpl/`
[9] `http://nlp.stanford.edu/software/corenlp.shtml`

Stanford CoreNLP tag MISC for all the other DBpedia classes (`Work`, `Event`, and so on). Finally, we connect each of these types to the corresponding DBpedia type/class. Boolean properties are not considered as they affect only 4 relations out of over 1,700 in the DBpedia ontology. See Table 2 for more information.

Table 2. Type conversion between Stanford NER and DBpedia

DBpedia	Stanford	DBpedia	Stanford
`Person`	PER	`date`	DATE
`Organisation`	ORG	`integer`	NUMBER
`Place`	LOC	`nonNegativeInteger`	NUMBER
`gYear`	DATE	`double`	NUMBER
`positiveInteger`	NUMBER	`[all other classes]`	MISC

5 Training

5.1 Retrieving sentences

Given an instance of a DBpedia relation, e.g. ⟨Barack Obama⟩ ⟨is_born_in⟩ ⟨Honolulu⟩, we examine the Wikipedia article text of the subject (i.e., "Barack Obama") looking for the two entities involved in the relation.

First, the sentences of the given article are pre-processed as described in Section 4 and we identify those sentences containing entities of the types involved in the relation. In the above example, the domain of ⟨is_born_in⟩ is `Person`, while its range is `Place`.

Then, we collect all the sentences containing entities classified both with domain and range types of the relation, where the corresponding strings of the triple match.

In the above example

> Obama was born on August 4, 1961 at Kapiolani Maternity & Gynecological Hospital in Honolulu, Hawaii, and is the first President to have been born in Hawaii

Stanford CoreNLP annotates the sentence in the following way:

> ⟨PER⟩Obama⟨/PER⟩ was born on ⟨DATE⟩August 4, 1961⟨/DATE⟩ at ⟨ORG⟩Kapiolani Maternity & Gynecological Hospital⟨/ORG⟩ in ⟨LOC⟩Honolulu⟨/LOC⟩, ⟨LOC⟩Hawaii⟨/LOC⟩, and is the first President to have been born in ⟨LOC⟩Hawaii⟨/LOC⟩.

The sentence contains both ⟨PER⟩ and ⟨LOC⟩, the conversion of domain and range type of relation ⟨is_born_in⟩ in DBpedia, respectively (see Section 4). Therefore it can be a candidate as a positive example for the relation. While there is a ⟨LOC⟩ part containing the range of the relation (Honolulu), the complete string of the domain (Barack Obama)

never appears in the sentence, so an approach based on exact string matching would erroneously discard this sentence. To avoid this behavior and increase the recall of this extraction step, we apply different matching strategies. First of all, we perform the exact match of the entire string as provided by Wikipedia. If the algorithm does not find such string in the sentence, we clean it deleting the part between brackets, used to disambiguate pages with the same title, as in "Carrie (novel)" and "Carrie (1976 film)". We use the resulting string for matching the domain in the sentence. If this fails, the original string is tokenized and, given the set of obtained tokens, new strings are built by combining the tokens (preserving the original word order). For instance, starting from "John Fitzgerald Kennedy", we obtain the new strings "John Fitzgerald", "John Kennedy", "Fitzgerald Kennedy", "John", "Fitzgerald" and "Kennedy". Using this rule, in our example we can identify the ⟨PER⟩ part containing the string "Obama".

For numeric entities, we do not use exact matching between the value stored in DBpedia and the number extracted by Stanford CoreNLP, as for some relations (such as `populationTotal`) they may be slightly different. Given two numeric values a and b, we then consider a positive match between them when a and b are both different from 0, and the ratio $|a - b| \,/\, |b|$ is less than 0.05 (5%).

5.2 Selecting sentences

Supervised machine learning tools need annotated data to be trained. Training (and test) sets consist of both *positive* and *negative* examples: the former are examples where the relation is present; the latter are examples where the relation is not expressed. The distant supervision paradigm uses structured data to collect positive examples, following the assumption that, if two entities participate in a relation, then all the sentences containing the two entities express such relation; however, this is not always true [19] (see below).

In our experiment, we use the hypothesis that a sentence containing the domain part of the relation, and not containing its range but another entity of the type of the range, is a good negative example for the relation ⟨is_born_in⟩. For example, the sentence

> Following high school, Obama moved to Los Angeles in 1979 to attend Occidental College.

contains "Obama", and does not contain "Honolulu" but contains "Los Angeles". Therefore we pick this sentence as a negative example for the relation.

This simple rule is sufficient for building a training set for relations where there is no ambiguity. For example, in a biographical article in Wikipedia the date of birth is usually used in the birth date relation only. In addition, other dates in the same sentences certainly refer to different relations, as the birth date of a person is unique.

However, there are relations where these assumptions are not necessarily true, since the same pair subject-object can be involved in more than one relation. Therefore, we apply different strategies for the extraction of the training data.

First strategy: positives cannot be negatives. In the sentence

> Obama was born on August 4, 1961 at Kapiolani Maternity & Gynecological Hospital in Honolulu, Hawaii, and is the first President to have been born in Hawaii.

the entity "Honolulu" refers to the birth place. Unfortunately, also the other ⟨LOC⟩ instance ("Hawaii") refers to the birth place (although it may not be included in the DBpedia resource). To avoid this problem, when collecting our training set we discard potential negative examples taken from a sentence already used to extract a positive one.

Second strategy: only one sentence per relation. In the sentence

> In 1971, Obama returned to Honolulu to live with his maternal grandparents, Madelyn and Stanley Dunham.

both "Obama" and "Honolulu" are present but the relation between them is different from birth place.

[19] tackle this problem by assuming that, if two entities participate in a relation, *at least one sentence* that mentions these two entities might express that relation. Using this assumption, they trained a graphical model with the optimization of the parameters to ensure that predictions will satisfy a set of user-defined constraints. In their work, they use the New York Times corpus, where pages are less standardized than in Wikipedia. In our experiment we can rely on a stronger assumption: if two entities participate in a relation, *there is one and only one sentence* expressing such relation. We can then discard those pages not complying with this assumption, i.e. having more than one sentence containing both domain and range of the relation.

Third strategy: only one relation for each value. Finally, we can take advantage from the complete set of properties available in DBpedia, by removing from the training set those pages having more than one relation sharing the same object. For instance, the two relations

$$\langle \text{Mark Zuckerberg} \rangle \ \langle \text{work\_for} \rangle \ \langle \text{Facebook} \rangle$$
$$\langle \text{Mark Zuckerberg} \rangle \ \langle \text{founded} \rangle \ \langle \text{Facebook} \rangle$$

involve the same pair subject-object, therefore we cannot disambiguate whether a sentence in Mark Zuckerberg's Wikipedia page containing both his name and the company he founded refers to the former or to the latter relation.

5.3 Training algorithm

As learning algorithm, we use jSRE, a state-of-the-art RE tool described in [8]. The RE task is treated as a classification problem in supervised learning, using kernel methods [21] to embed the input data into a suitable feature space, and then run a classical linear algorithm to discover nonlinear patterns. The learning algorithm used is Support Vector Machines (SVM) [25, 6].

The tool uses two families of kernels: *global context kernel* and *local context kernel*. The first one adapts the ideas in [4] and uses a bag-of-words of three sets of tokens: fore-between (tokens before and between the two entities), between (only tokens between the two entities), and between-after (tokens between and after the two entities). The second kernel represents the local context using NLP basic features such as: lemma, part-of-speech, stem, capitalization, punctuation, and so on.

6 Experiments and evaluation

We choose seven different DBpedia properties to evaluate the system, covering different domain-range pairs. For each relation, we extract 50,000 pages for training, 1,000 for development and 1,000 for test (except for the `capital` property, for which not enough instances are available). All the experiments are conducted on a test set built following the same steps used for training.

The strategy used to calculate precision and recall is *One Answer per Slot* [12]:

- if the system finds in the document the correct value for the desired relation, we do not count any false negative for the same relation;
- if the system does not find it (or the value is wrong), we count it as a false negative;
- false positives are not affected by this method, therefore they are all counted when computing precision and recall.

Table 3 shows the results obtained on the seven considered relations, with and without the application of the strategies described in Section 5.2. The second column reports the number of occurrences in DBpedia for the relation. The property `capital` is not very frequent, but we choose to use it in our test to demonstrate that our approach is also feasible when we do not have a large number of examples. Unfortunately, in this case the three strategies (see Section 5.2) are not applicable, because the number of examples (positive and negative) drops due to the required constraints .

Table 3. Evaluation of the system on seven DBpedia properties

Property	Instances	Without strategies			With strategies		
		P	R	F_1	P	R	F_1
`birthDate`	258,609	0.92	1.00	0.95	0.91	1.00	0.95
`capital`	3,921	0.82	0.67	0.74	-	-	-
`deathDate`	82,060	0.94	0.99	0.96	0.93	1.00	0.96
`headquarter`	27,283	0.71	0.84	0.77	0.81	0.80	0.80
`populationTotal`	237,701	0.68	0.86	0.76	0.70	0.87	0.78
`region`	66,269	0.87	0.90	0.88	0.91	0.91	0.91
`deathPlace`	123,705	0.51	0.61	0.56	0.59	0.77	0.67

The results show that the application of the three strategies increases the F_1 value. In some cases (`birthDate`, `deathDate`, `populationTotal`) the increase is neg-

ligible, because the corresponding relations are not affected by the issue described in [19].

7 Conclusions and future work

This paper proposes a method to extract missing DBpedia properties from the article text of Wikipedia pages. The approach is based on the distant supervision paradigm, and makes use of supervised machine learning for the extraction.

The accuracy of our approach is comparable to other systems'. However, a precise comparison is hard to make, because they are applied on different resources and tasks. In [15], Yago is used as resource to collect training sentences, while [24] uses DBpedia and the distant supervision paradigm for the TAC-KBP slot filling task.

Due to the high variability and complexity of the task, much work is still to be done, and different issues should be addressed:

- Disambiguation tools and Wikipedia links could be used for sentence retrieving (see Section 5.1).
- In our experiments we have used jSRE as an out-off-the-shelf tool. We plan to investigate the use of kernels exploiting Wikipedia-related features, such as internal links.
- To increase the number of sentences that can be used for training, some approaches (e.g., [24]) use shallow coreference resolution using animate pronouns,. In real world applications, where the number of relations is high and the number of examples is not, a more sophisticated coreference resolution tool can help to obtain more training data.
- Distant supervision is a language-independent paradigm, although most of the resources and approaches concerns only English, and the multi-linguality of the approach has not been deeply investigated. DBpedia releases its resource in 16 languages, therefore it can be in principle used to apply distant supervision on languages for which suitable natural language tools are available (such as TextPro[10], OpenNLP[11] or Stanbol[12]). There is a preliminary work on applying distant supervision on the Portuguese Wikipedia and DBpedia [2].

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Proceedings of the 6th international Semantic Web Conference and 2nd Asian Semantic Web Conference. pp. 722–735. ISWC'07/ASWC'07, Springer-Verlag, Berlin, Heidelberg (2007), http://dl.acm.org/citation.cfm?id=1785162.1785216

[10] http://textpro.fbk.eu/
[11] http://opennlp.apache.org/
[12] http://stanbol.apache.org/

2. Batista, D.S., Forte, D., Silva, R., Martins, B., Silva, M.: Extracção de Relações Semânticas de Textos em Português Explorando a DBpédia e a Wikipédia. Linguamatica 5(1), 41–57 (Julho 2013), http://www.linguamatica.com/index.php/linguamatica/article/view/157
3. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data. pp. 1247–1250. SIGMOD '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1376616.1376746
4. Bunescu, R.C., Mooney, R.J.: Subsequence kernels for relation extraction. In: NIPS (2005)
5. Cabrio, E., Cojan, J., Palmero Aprosio, A., Magnini, B., Lavelli, A., Gandon, F.: QAKiS: an open domain QA system based on relational patterns. In: Glimm, B., Huynh, D. (eds.) International Semantic Web Conference (Posters & Demos). CEUR Workshop Proceedings, vol. 914. CEUR-WS.org (2012), http://dblp.uni-trier.de/db/conf/semweb/iswc2012p.html#CabrioCAMLG12
6. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press (2010)
7. Exner, P., Nugues, P.: Entity extraction: From unstructured text to DBpedia RDF triples. In: Proceedings of the Web of Linked Entities Workshop in conjuction with the 11th International Semantic Web Conference. pp. 58–69. Boston (2012), http://ceur-ws.org/Vol-906/paper7.pdf
8. Giuliano, C., Lavelli, A., Romano, L.: Relation extraction and the influence of automatic named-entity recognition. ACM Transactions on Speech and Language Processing 5(1), 2:1–2:26 (Dec 2007), http://doi.acm.org/10.1145/1322391.1322393
9. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. Processing pp. 1–6 (2009), http://www.stanford.edu/~alecmgo/papers/TwitterDistantSupervision09.pdf
10. Krause, S., Li, H., Uszkoreit, H., Xu, F.: Large-scale learning of relation-extraction rules with distant supervision from the web. In: Cudr-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J., Hendler, J., Schreiber, G., Bernstein, A., Blomqvist, E. (eds.) The Semantic Web ISWC 2012, Lecture Notes in Computer Science, vol. 7649, pp. 263–278. Springer Berlin Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-35176-1_17
11. Lange, D., Böhm, C., Naumann, F.: Extracting structured information from Wikipedia articles to populate infoboxes. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management. pp. 1661–1664. CIKM '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1871437.1871698
12. Lavelli, A., Califf, M., Ciravegna, F., Freitag, D., Giuliano, C., Kushmerick, N., Romano, L., Ireson, N.: Evaluation of machine learning-based information extraction algorithms: criticisms and recommendations. Language Resources and Evaluation 42(4), 361–393 (2008), http://dx.doi.org/10.1007/s10579-008-9079-3
13. Lehmann, J., Gerber, D., Morsey, M., Ngomo, A.C.N.: DeFacto - Deep fact validation. In: International Semantic Web Conference (1). pp. 312–327 (2012)
14. 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
15. Nguyen, T.V.T., Moschitti, A.: End-to-end relation extraction using distant supervision from external semantic repositories. In: ACL (Short Papers). pp. 277–282 (2011)

16. Palmero Aprosio, A., Giuliano, C., Lavelli, A.: Automatic expansion of DBpedia exploiting Wikipedia cross-language information. In: Proceedings of the 10th Extended Semantic Web Conference (2013)
17. Palmero Aprosio, A., Giuliano, C., Lavelli, A.: Automatic Mapping of Wikipedia Templates for Fast Deployment of Localised DBpedia Datasets. In: Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies (2013)
18. Palmero Aprosio, A., Giuliano, C., Lavelli, A.: Towards an Automatic Creation of Localized Versions of DBpedia. In: Proceedings of the 12th International Semantic Web Conference (2013)
19. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part III. pp. 148–163. ECML PKDD'10, Springer-Verlag, Berlin, Heidelberg (2010), http://dl.acm.org/citation.cfm?id=1889788.1889799
20. Roth, B., Barth, T., Wiegand, M., Klakow, D.: A survey of noise reduction methods for distant supervision. In: Automated Knowledge Base Construction 2013. Proceedings of the 3rd Workshop on Knowledge Extraction at CIKM 2013. California, USA (2013)
21. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
22. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web. pp. 697–706. WWW '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1242572.1242667
23. Sultana, A., Hasan, Q.M., Biswas, A.K., Das, S., Rahman, H., Ding, C., Li, C.: Infobox suggestion for Wikipedia entities. In: Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 2307–2310. CIKM '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2396761.2398627
24. Surdeanu, M., McClosky, D., Tibshirani, J., Bauer, J., Chang, A.X., Spitkovsky, V.I., Manning, C.D.: A simple distant supervision approach for the TAC-KBP slot filling task. In: Proceedings of the Third Text Analysis Conference (TAC 2010). Gaithersburg, Maryland, USA (November 2010), pubs/kbp2010-slotfilling.pdf
25. Vapnik, V.: An overview of statistical learning theory. IEEE Transactions on Neural Networks 10(5), 988–999 (1999)
26. Vrandečić, D.: Wikidata: a new platform for collaborative data collection. In: Proceedings of the 21st international conference companion on World Wide Web. pp. 1063–1064. WWW '12 Companion, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2187980.2188242