Named Entity Extraction in Requirement Specification: A Comparison

Luciana Tanevitch[1][0000−0002−5322−9314], Leandro Antonelli[1][0000−0003−1388−0337], and Diego Torres[1,2][0000−0001−7533−0133]

[1] LIFIA, CICPBA-Facultad de Informática, UNLP, Argentina
{name.surname}@lifia.info.unlp.edu.ar
[2] Departamento de Ciencia y Tecnología, UNQ, Argentina

Abstract Software requirements specifications generally are written in natural language. Identifying and extracting the main concepts involved in a requirements specification could be useful for the development process, quality assurance, and software maintenance. However, a computer agent is not able to process and understand immediately the content and information included in the natural language documents. Named entity extraction is a task that involves recognizing entities in a text and linking them to a knowledge graph to disambiguate them. In the field of requirements, applying this task can be useful for building structures that allow for the representation and efficient management of complex information. Different tools are focused in entity extraction and then an entity linking with a specific knowledge graph such as Wikidata. This work compares different named entity extraction tools in the task of extracting entities in a requirements specification.

Keywords: Named Entity Extraction · Knowledge graph · Requirements engineering

1 Introduction

Software requirements specifications (SRS) define in an unstructured way the requirements that a system should satisfy. Identifying and extracting the main concepts involved in a requirements specification could be useful for the development process, quality assurance, and software maintenance. They are also useful for document classification tasks, grouping SRS by their purpose or content similarities [17]. These texts are mainly in natural language, which allows humans to understand and exchange information. However, a computer agent is not able to process and understand immediately the content and information included in the natural language documents. Indeed, research lines introduced techniques are necessary to enable machines to convert text into information that can be processed automatically and to deal with the ambiguity of natural language [3,7].

Natural language processing (NLP) is a branch of the Artificial Intelligence that enables computers to understand texts written in natural language[15,18]. NLP can be used to extract entities from a text using a specific technique called

named entity extraction (NER), which allows recognition and classification of named entities in a text into predefined classes [15]. For example, in the sentence *"Google is a widely used search engine"*, the mention *Google* could be categorized as an *Organization*. As natural language allows for multiple meanings of the same concept, once entities are detected, it is necessary to disambiguate them to determine their true meaning according to the context in which they occur.

Knowledge graphs (KG) allow structuring complex information in a proper format for computers, and they can be specified in languages such as OWL and RDF [13]. Each node of the graph must represent a unique concept, which is achieved by using Internationalized Resource Identifiers (IRI). An IRI uniquely identifies a resource on the Web. These IRIs can be obtained from vocabularies or ontologies existing in Linked Open Data (LOD)[28], which aims to define the meaning of a concept in the most accurate way. Linking a named entity in a text to its corresponding IRI enables meaning disambiguation. Knowledge graphs are suitable for automatic data processing.

KGs can be built from texts written in natural language by using NLP techniques. Detecting entities is a relevant task in KG construction as they are nodes of the graph. The process from extracting entities to represent them as nodes of a KG is known as Named Entity Extraction (NEE) and it involves three main tasks: named entity recognition (NER), named entity disambiguation (NED) and named entity linking (NEL). To clarify, NER is the task of identifying and classifying named entities in text (e.g. identifying "Google" as an organization), NED is the task of determining the correct identity for an entity (e.g. disambiguating "Apple" as the company vs. the fruit), and NEL is the task of linking entities in a text to their corresponding nodes in a knowledge graph. There are many approaches for each of these stages, and many tools that implement them. Al-Moslmi et. al propose a pipeline for transforming texts in KGs [2]. To choose a tool for applying on the context of requirement analysis one must take into account different core features of the tools. This work considers the supported language, disambiguation techniques and the knowledge graph used for linking entities.

The aim of this paper is to compare various NEE tools applied to the domain of requirements, taking into account the aforementioned core features and their performance (measured in terms of precision and recall) in requirements specification analysis.

This paper is organized as follows. In Section 2, previous work in the literature in this area is introduced. Then, Section 3 defines the evaluation method and also it includes the metrics, the tools to be compared, the data used for the evaluation. The results of the evaluation are described in Section 4. Finally, Section 5 summarizes the paper conclusions and suggests some possible future work.

2 Related Work

Tedeschi et. al [24] evaluate various NER-based strategies which allow systems trained on limited amounts of data to narrow the performance gap with those systems trained on massive training corpora. Vychegzhanin and Kotelnikov[27] conducted a comparison of domain-independent NER tools, considering their characteristics and performance in recognizing persons, organizations, places, and time indicators on pre-trained news datasets. Checco et. al develop a tool that detects named entities in fashion blogs and links them to their own ontology about fashion concepts to discover new items or trends. [6]. Hosseni and Bagheri[14] consider the application of different entity linking techniques for detecting implicit entities in a text (in particular tweets due to frequently there are implicit references on them), and categorize features that can be used for explicit and implicit entity linking within the context of a learning to rank approach. Rizzo y Troncy [22] propose a framework for people to evaluate some NEE tools based on three criteria: named entity detection, entity type detection, and entity disambiguation. Foppiano and Romary [12] present a tool with a user interface that enables to automatically extract entities from a document, and visualize an infobox about each concept.

3 Evaluation Method

To compare the different NEE tools, an evaluation was designed. The main questions that structure the evaluation are the followings:

- What are the supported languages?
- How does the approach disambiguate name entities?
- Which is the external knowledge base used as a reference?
- How much accurate is the approach?

The following section describes the characteristics and metrics used to answer the former questions, then, the list of evaluated tools, the data input for these tools, and the evaluation results.

3.1 Metrics

Core features As it was mentioned, three elements should be considered for evaluating the tools' core features: supported language, disambiguation technique, and knowledge graph used.

The "language" feature lists the languages in which the input text can be written and the tool can understand them. Language is important in the multicultural requirements description because specific terminology is hard to be translated into an intermediate language, such as English.

The "disambiguation technique" aspect briefly describes the mechanism used by the tool to disambiguate named entities based on the context in which they

appear. They are a wide range of techniques that could include from the use of machine learning to the computation of the Term frequency–inverse document frequency (TF-IDF) algorithm.

Finally, the knowledge base that the tool has for entity assignment. DBpedia [4,16] and Wikidata [26,10] are the most relevant knowledge bases used for entity linking, especially for entities that are in the domain of the real world.

Performance Four parameters are used for evaluating the performance of the tool: precision, recall, f1-score and accuracy. The precision metric allows us to determine the model's ability to correctly identify entities. It is calculated using the formula 1, where TP represents the true positives (i.e., entities correctly recognized by the tool) and FP represents the false positives (i.e., entities wrongly identified by the tool). Recall measures the ability of the model to identify all correct entities and is calculated as 2, where TP represents true positives and FN represents false negatives (i.e., entities that should be recognized by the tool, but were not). F1-score is an harmonic measure between precision and recall, and it is calculated as 3.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{F1-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3}$$

Accuracy allows us to determine if the resource linked to the named entity is correct for the context in which the entity occurs. For example, in the sentence *Apple is an American brand*, *Apple* should be linked to a resource that refers to the brand and not the fruit. Accuracy is calculated by dividing the number of entities correctly linked to a knowledge graph by the total number of entities.

3.2 Data

The tools considered in this work are domain-independent, i.e, they are able to recognize entities in texts of any topic. Wikifier [5] is an implementation of an approach named *Wikification*, that is using Wikipedia as a general-purpose ontology for linking named entities, so it is possible to annotate documents in any of the languages in which Wikipedia is available. DBpedia Spotlight [19] is a tool for detecting DBpedia resources in texts, allowing it to be configured based on prominence, contextual ambiguity, topical pertinence and confidence scores. Babelfy [20] uses BabelNet[21] as a knowledge graph, which supports multiple languages and integrates lexicographic and encyclopedic knowledge. This allows them to apply Word Sense Disambiguation (WSD) and Entity Linking (EL) approaches together. TagMe [11] is a system that is able to efficiently and judiciously augment a plain-text with pertinent hyperlinks to Wikipedia pages.

Spacy is an NLP library that includes an entity linking module which can be configured to support various knowledge graphs.

Many of these tools have user interfaces that facilitate the user experience for testing entity disambiguation. An example of these interfaces is shown in Figure 1

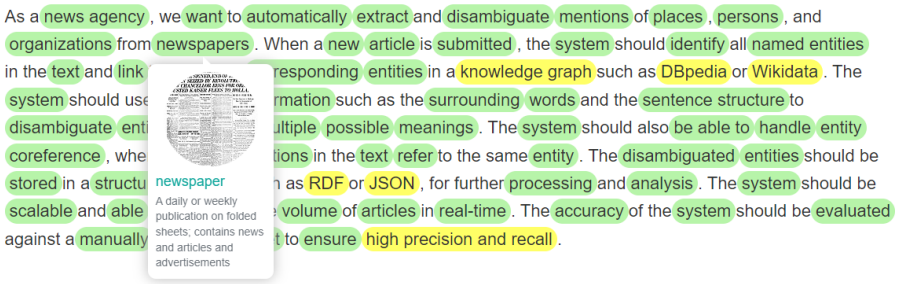

Figure 1. Babelfy's user interface

The text in Table 1 was used as input for the tools. The specification was written by a functional analyst, and it describes a functional requirement that involves various technical concepts related to the named entity extraction pipeline. An expert analysis can be performed on this data to determine which entities should be automatically detected. It is called *ground truth* dataset because it contains the entities expected to be found by the evaluated tools, and it is defined at Table 2.

As a news agency, we want to automatically extract and disambiguate mentions of places, persons, and organizations from newspapers. When a new article is submitted, the system should identify all named entities in the text and link them to their corresponding entities in a knowledge graph such as DBpedia or Wikidata. The system should use contextual information such as the surrounding words and the sentence structure to disambiguate entities that have multiple possible meanings. The system should also be able to handle entity coreference, where different mentions in the text refer to the same entity. The disambiguated entities should be stored in a structured format, such as RDF or JSON, for further processing and analysis. The system should be scalable and able to handle a large volume of articles in real-time. The accuracy of the system should be evaluated against a manually labeled dataset to ensure high precision and recall.

Table 1. Functional requirements specification

Considering Wikidata knowledge graph as example, we can discuss how should tools disambiguate some named entities. The concept *system* should be

news agency, places, person, organization, article, system, named entity, newspapers, knowledge graph, DBpedia, Wikidata, contextual information, sentence structure, meanings, coreference, RDF, JSON, dataset, accuracy, analysis, precision, recall

Table 2. Ground Truth

ideally linked to entity Q2429814 of Wikidata, which represents a software system, as the text is describing the features that a software has to achieve. It could be admissible, although less precise, to be linked to entity Q58778 which represents a system (not necessary related to IT).

As *news agency* and *knowledge graph* are compound words, they should be considered as a single concept. So Q192283 is a right entity for news agency, and Q33002955 for knowledge graph. Moreover, *meanings* could be linked to semantics (Q39645) and *sentence structure* to syntax (Q37437). *Entity* must be disambiguated in the context of entity recognition. It would be wrong to bind it to the context of entity-relationship model, for example.

The full table of expected entities is shown in Table 5. The table lists each mention that must be found in the text along with its expected entity. Due to space constraints, each row in the table contains two different mentions. There are three types of values in the "expected entity" column, each corresponding to a resource on a knowledge graph. The prefix of each value indicates which knowledge graph it belongs to: "wd" for Wikidata, "dbr" for DBpedia, and "bn" for BabelNet. Because each knowledge base has its own resources, we must search for the expected entities in each knowledge graph that a tool can use to define the "best" entity that matches a given mention. Additionally, there may be some concepts that are related to the best match according to human criteria, and these could also be considered tolerable entities for a given mention.

4 Results

Table 4 details the comparison of the core features of the tools mentioned in Section 3. We can see that all of these tools support multiple languages, with English being the common language among them. Furthermore, not all tools support the same number of languages, with Babelfy supporting the most. Given that BabelNet, DBpedia, and Wikidata are multilingual knowledge bases, we can link mentions found in texts written in different languages to them. Therefore, a supported language implies that the tool must have an NLP module capable of processing text in a specific language, as well as a knowledge base that contains entries in that language. This can mean that a tool is pretty good for a English but not for Spanish, everything depends on the NLP module and the amount of data contained in the used KG in that language.

Most of the tools focus on machine learning approaches such as word embeddings for the disambiguation phase, regardless of the technique used for candidate

Tool	Language	Disambiguation technique	Knowledge graph
DBpedia Spotlight	Among the supported languages are German, English, Spanish, French, Italian, and Portuguese. The full list can be found at [9]	They modeled DBpedia resource occurrences in a Vector Space Model (VSM) using a variant of the TF-IDF algorithm to weigh words based on their ability to distinguish between candidates for a given surface form. Therefore, they use cosine similarity to compare the similarity between context vectors and the context surrounding the surface form. In an improved version [8], they use a generative model to calculate the probability that a candidate is correct for a mention, which improves disambiguation accuracy, as well as time performance and required space	DBpedia
Wikifier	It supports around 100 languages among which are German, English, Spanish, French, Italian, Portuguese, and Chinese.	They build a bipartite graph where each node on the left represents mentions and each node on the right represents Wikipedia entries for those mentions. The graph is augmented by edges between concepts based on their semantic relatedness. They use the graph to calculate the PageRank score for each vertex, and after some iterations, they obtain the relevant concepts for each mention. A threshold value can be specified by the user to discard all candidates that were scored below that value.	The entities are linked to wikipedia pages, although the service includes information from the Wikidata ID and DBpedia URI in the response.
Babelfy	Provides support for 271 languages [1], including English	They build a directed graph which relates mentions in the text with their Babel candidate They construct a graph that relates mentions in the text to their possible candidates in Babel, also linking those candidates that are semantically similar. In this way, all possible interpretations are obtained for each mention, and the heuristic of the densest subgraph is applied to obtain the best candidate for a mention (based on lexical and semantic coherence), and thus arrive at the most coherent semantic interpretation.	BabelNet
TagMe	English, Italian and German	They apply a voting scheme where the goal is to reach a "collective agreement" on the real meaning of some mentions. To select the best candidate, they tested two algorithms: Disambiguation by Classifier (DC), which computes the probability of correct disambiguation for every candidate of a mention and then selects the best one; and Disambiguation by Threshold (DT), which selects the top n-best candidates. Then other works improved that initial version by topical classification and clustering [25] [23].	Entities are linked to a Wikipedia page
Spacy	English, although a model can be trained to recognize other languages	Starting from a defined knowledge base, candidates for entities can be generated. Then, a machine learning model selects the most suitable candidate. The power of Spacy lies in the fact that each component is customizable, allowing the user to improve accuracy by combining techniques for candidate generation and scoring.	It allows configuring the desired knowledge base, although in this work, an implementation with Wikidata was used.

Table 4. Core features comparison results

Entity mention	**Expected entity**	**Entity mention**	**Expected entity**
news agency	wd:Q192283 dbr:News_agency bn:00057550n	article	wd:Q191067 dbr:Article_(publishing) bn:00006121n
system	wd:Q2429814 dbr:System_software bn:00021497n	newspapers	wd:Q11032 dbr:Newspaper bn:00057563n
place	wd:Q2221906 dbr:Location bn:00062699n	person	wd:Q215627 dbr:Person bn:00046516n
organization	wd:Q43229 dbr:Organization bn:00059480n	knowledge graph	wd:Q33002955 dbr:Knowledge_graph bn:02930995n
named entity	wd:Q25047676 dbr:Named_entity bn:17388870n	DBpedia	wd:Q465 dbr:DBpedia bn:00322825n
Wikidata	wd:Q2013 dbr:Wikidata bn:02886551n	contextual information	wd:Q196626 dbr:Context bn:00022168n
sentence structure	wd:Q37437 dbr:Syntax bn:00062118n	meaning	wd:Q183046 dbr:Semantics bn:00046139n
coreference	wd:Q63087 dbr:Coreference bn:00022637n	RDF	wd:Q54872 dbr:RDF bn:03335263n
JSON	wd:Q2063 dbr:JSON bn:00802141n	dataset	wd:Q1172284 dbr:Data_set bn:03731507n
accuracy	wd:Q272035 dbr:Accuracy_and_precision bn:00000782n	analysis	wd:Q1988917 dbr:Data_analysis bn:02081302n
precision	wd:Q2359161 dbr:Precision_and_recall bn:00641989n	recall	wd:Q2359161 dbr:Precision_and_recall bn:00641989n

Table 5. Expected mention linking

selection. Word embedding allows to represent a concepts as numeric vectors. So there can be applied some similarity measures (as cosine, for example) for comparing the representation of the named entity with their candidate entities. While some tools use word embeddings to take the context of the surface form into account, others take advantage of the graph structure of the data and analyse the semantic relationships among the entities to determine the best candidates.

Evaluated tools can link mentions to Wikidata, DBpedia or BabelNet. Although they store a large amount of information and they are widely used, there are others existing knowledge bases as YAGO and Freebase. All of them contain general concepts of the natural language, but not in a specific domain as requirements. So there are some concepts that are not going to be recognized because of the domain, but that doesn't mean that these knowledge graphs are bad for other domains. The importance of BabelNet is that it includes lexical resources, which provide a foundation of structured knowledge, so using lexical and semantic knowledge could improve word sense disambiguation tasks.

Tool	**Accuracy**	**Precision**	**Recall**	**F1-score**
DBpedia Spotlight	1	1	0.27	0.42
Wikifier	0.85	0.66	0.63	0.65
Babelfy	0.52	0.39	0.95	0.55
TagMe	**0.73**	**0.71**	**0.9**	**0.80**
Spacy	0.68	0.66	0.9	0.76

Table 6. Performance evaluation results

Table 6 shows the comparison of the core features of the tools. Here we can see that even if DBpedia Spotlight has a perfect precision and accuracy score, it has a low recall, i.e., it recognized a low amount of entities compared to those existing in the ground truth. On the other hand, Babelfy recognized almost all the words defined in the ground truth, but it had a low precision due to overfitting, i.e., it detected more entities than those contained in the ground truth. This could have two causes: either the tool is recognizing unnecessary concepts, or there are some concepts that the domain expert left out of the ground truth, which should be considered.

TagMe and Spacy kept a good balance in all their measures: they recognized almost all the expected words with a precision higher than 65%.

Finally, Wikifier had a good performance for detecting entities, but it had a lower recall compared to the tools mentioned above. It's worth mentioning that when a threshold configuration is available for the tool, that value was set as an average 0.5. This is useful for balancing performance, because a low threshold value may include too many wrong results (penalizing precision), but a high threshold value may recognize too few of the mentions (affecting recall).

TagMe obtained the higher results for the used data, but Spacy was so close: it has similar values for accuracy and precision, keeping the same recall. So we can say that even if TagMe had a good performance in requirement domain, better results could be obtained using an specific-domain knowledge graph for training Spacy.

5 Conclusions

The domain of requirements is highly ambiguous and complex. In order to enable automatic processing of requirement specifications, techniques such as knowledge graphs can be employed. Linking a concept to a knowledge base allows for disambiguation of its meaning, and entity extraction tools can be used for this purpose. Evaluation results exhibit considerable variability in accuracy, precision, and recall. Some tools demonstrate high precision but low recall, indicating their ability to accurately recognize entities but with the potential to overlook others. While precision and recall are useful metrics, accuracy is critical in determining whether an entity disambiguation tool is performing correctly. In all the evaluated tools, accuracy value was over precision. Also we have to keep in mind that these tools could be useful for detecting entities that are out of the ground truth dataset, helping domain experts to improving their analysis.

It is possible that the combination of different tools can improve results and provide greater accuracy in identifying requirement entities. Tools such as Spacy, which allow customization of the main components of the process, can be advantageous in applying various techniques for generating and scoring candidates, as well as tailoring their training to the requirements domain.

This evaluation may be applied to different domains to determine whether the tools have the same ability to recognize entities as they do in the requirements domain. The next step could be to detect relationships between the identified entities in order to build a knowledge graph.

References

1. (Apr 2015), http://babelfy.org/javadoc/it/uniroma1/lcl/jlt/util/Language.html
2. Al-Moslmi, T., Gallofré Ocaña, M., Opdahl, A., Veres, C.: Named Entity Extraction for Knowledge Graphs: A Literature Overview. IEEE Access **8**, 32862–32881 (Feb 2020). https://doi.org/10.1109/ACCESS.2020.2973928
3. Antonelli, L., Delle Ville, J., Dioguardi, F., Fernández, A., Tanevitch, L., Torres, D.: An iterative and collaborative approach to specify scenarios using natural language. In: Workshop on Requirements Engineering (WER22),(Modalidad virtual, 23 al 26 de agosto de 2022) (2022)
4. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. In: The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007+ ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings. pp. 722–735. Springer (2007)
5. Brank, J., Leban, G., Grobelnik, M.: Annotating documents with relevant wikipedia concepts. Proceedings of SiKDD **472** (2017)

6. Checco, A., Demartini, G., Löser, A., Arous, I., Khayati, M., Dantone, M., Koopmanschap, R., Stalinov, S., Kersten, M., Zhang, Y.: Fashionbrain project: A vision for understanding europe's fashion data universe. arXiv preprint arXiv:1710.09788 (2017)
7. Corral, A., Sánchez Crespo, L.E., Antonelli, L.: Building an integrated requirements engineering process based on intelligent systems and semantic reasoning on the basis of a systematic analysis of existing proposals. JUCS - Journal of Universal Computer Science **28**, 1136–1168 (11 2022). https://doi.org/10.3897/jucs.78776
8. Daiber, J., Jakob, M., Hokamp, C., Mendes, P.N.: Improving efficiency and accuracy in multilingual entity extraction. In: Proceedings of the 9th International Conference on Semantic Systems. pp. 121–124. ACM, Graz Austria (Sep 2013). https://doi.org/10.1145/2506182.2506198, https://dl.acm.org/doi/10.1145/2506182.2506198
9. DBpedia Spotlight - Shedding light on the web of documents: FAQ. https://www.dbpedia-spotlight.org/faq (accessed March 8, 2023)
10. Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., Vrandečić, D.: Introducing wikidata to the linked data web. In: The Semantic Web–ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I 13. pp. 50–65. Springer (2014)
11. Ferragina, P., Scaiella, U.: 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. pp. 1625–1628. ACM, Toronto ON Canada (Oct 2010). https://doi.org/10.1145/1871437.1871689, https://dl.acm.org/doi/10.1145/1871437.1871689
12. Foppiano, L., Romary, L.: entity-fishing: a dariah entity recognition and disambiguation service. Journal of the Japanese Association for Digital Humanities **5**(1), 22–60 (2020)
13. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutierrez, C., Gayo, J.E.L., Kirrane, S., Neumaier, S., Polleres, A., Navigli, R., Ngomo, A.C.N., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., Zimmermann, A.: Knowledge Graphs. ACM Computing Surveys **54**(4), 1–37 (May 2022). https://doi.org/10.1145/3447772, http://arxiv.org/abs/2003.02320, arXiv:2003.02320 [cs]
14. Hosseini, H., Bagheri, E.: From Explicit to Implicit Entity Linking: A Learn to Rank Framework. In: Goutte, C., Zhu, X. (eds.) Advances in Artificial Intelligence, vol. 12109, pp. 283–289. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-47358-7_28, http://link.springer.com/10.1007/978-3-030-47358-7_28, series Title: Lecture Notes in Computer Science
15. Khurana, D., Koli, A., Khatter, K., Singh, S.: Natural language processing: state of the art, current trends and challenges. Multimedia Tools and Applications **82**(3), 3713–3744 (Jan 2023). https://doi.org/10.1007/s11042-022-13428-4, https://link.springer.com/10.1007/s11042-022-13428-4
16. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al.: Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web **6**(2), 167–195 (2015)
17. Malik, G., Cevik, M., Khedr, Y., Parikh, D., Başar, A.: Named Entity Recognition on Software Requirements Specification Documents. Proceedings of the Canadian Conference on Artificial Intelligence (Jun 2021). https://doi.org/10.21428/594757db.507e7951, https://caiac.pubpub.org/pub/v0la26hg

18. Martinez, A.R.: Natural language processing. Wiley Interdisciplinary Reviews: Computational Statistics **2**(3), 352–357 (2010)
19. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems. pp. 1–8. ACM, Graz Austria (Sep 2011). https://doi.org/10.1145/2063518.2063519, https://dl.acm.org/doi/10.1145/2063518.2063519
20. Moro, A., Raganato, A., Navigli, R.: Entity Linking meets Word Sense Disambiguation: a Unified Approach. Transactions of the Association for Computational Linguistics **2**, 231–244 (Dec 2014). https://doi.org/10.1162/tacl_a_00179, https://direct.mit.edu/tacl/article/43316
21. Navigli, R., Ponzetto, S.P.: Babelnet: Building a very large multilingual semantic network. In: Proceedings of the 48th annual meeting of the association for computational linguistics. pp. 216–225 (2010)
22. Rizzo, G., Troncy, R.: NERD: Evaluating Named Entity Recognition Tools in the Web of Data (2011)
23. Scaiella, U., Ferragina, P., Marino, A., Ciaramita, M.: Topical clustering of search results. In: Proceedings of the fifth ACM international conference on Web search and data mining. pp. 223–232. ACM, Seattle Washington USA (Feb 2012). https://doi.org/10.1145/2124295.2124324, https://dl.acm.org/doi/10.1145/2124295.2124324
24. Tedeschi, S., Conia, S., Cecconi, F., Navigli, R.: Named Entity Recognition for Entity Linking: What Works and What's Next. In: Findings of the Association for Computational Linguistics: EMNLP 2021. pp. 2584–2596. Association for Computational Linguistics, Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.findings-emnlp.220, https://aclanthology.org/2021.findings-emnlp.220
25. Vitale, D., Ferragina, P., Scaiella, U.: Classification of Short Texts by Deploying Topical Annotations. In: Baeza-Yates, R., de Vries, A.P., Zaragoza, H., Cambazoglu, B.B., Murdock, V., Lempel, R., Silvestri, F. (eds.) Advances in Information Retrieval. pp. 376–387. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28997-2_32
26. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM **57**(10), 78–85 (2014)
27. Vychegzhanin, S., Kotelnikov, E.: Comparison of Named Entity Recognition Tools Applied to News Articles. In: 2019 Ivannikov Ispras Open Conference (ISPRAS). pp. 72–77 (Dec 2019). https://doi.org/10.1109/ISPRAS47671.2019.00017
28. Yu, L., Yu, L.: Linked open data. A Developer's Guide to the Semantic Web pp. 409–466 (2011)