Leveraging Wikipedia Navigational Templates for Curating Domain-Specific Fuzzy Conceptual Bases

Krati Saxena, Tushita Singh, Ashwini Patil, Sagar Sunkle, Vinay Kulkarni
Tata Consultancy Services Research
Pune, India

Abstract

Domain-specific conceptual bases use key concepts to capture domain scope and relevant information. Conceptual bases serve as a foundation for various downstream tasks, including ontology construction, information mapping, and analysis. However, building conceptual bases necessitates domain awareness and takes time. Wikipedia navigational templates offer multiple articles on the same/similar domain. It is possible to use the templates to recognize fundamental concepts that shape the domain. Earlier work in this domain used Wikipedia's structured and unstructured data to construct open-domain ontologies, domain terminologies, and knowledge bases. We present a novel method for leveraging navigational templates to create domain-specific fuzzy conceptual bases in this work. Our system generates knowledge graphs from the articles mentioned in the template, which we then process using Wikidata and machine learning algorithms. We filter important concepts using fuzzy logic on network metrics to create a crude conceptual base. Finally, the expert helps by refining the conceptual base. We demonstrate our system using an example of RNA virus antiviral drugs.

1 Introduction

Domain-specific conceptual bases are a method for grasping the domain at a high level by capturing the notions that generally make up a domain. While ontology focus on formal representations and system of categories encompassing the domain information and conceptual models focus on linking the general ontological categories (Fonseca and Martin, 2007), the conceptual bases are abstract models addressing the most crucial concepts that are invariably found in a domain. Aside from defining the scope and outlining the concepts, the conceptual bases may be used for a variety of downstream activities, such as developing less abstract conceptual constructs, such as ontology, or applications such as entity mapping in knowledge graphs, creating instances for named entity recognition, and summarizing or analyzing the domain.

Creating a conceptual base is a difficult task that necessitates a thorough understanding of the domain and a considerable amount of time to establish the importance of concepts. Online sources such as Wikipedia contain a vast amount of information on many domains (Wikipedia, 2021a). In this research, we propose a novel approach to create domain-specific conceptual bases using Wikipedia navigational templates (Wikipedia, 2021b). The navigational templates make it simple to connect similar topics invariably. Similar topics are present as navigational boxes at the bottom of the article or sidebars on the right side of the article.

Our system uses knowledge from the articles in the navigational templates and identifies relevant notions consistently present in various articles of the same field. For this, we parse the articles' information and create a basic knowledge graph. We map the information to their Wikidata instances and cluster similar concepts. We apply fuzzy rules based on network metrics to decide the importance of concepts. In the end, the expert cleans and refines the resultant conceptual base to create the final version.

Our specific contributions are:

- Our framework allows users to build domain-specific conceptual bases from knowledge graphs in various domains using Wikipedia navigational templates.
- The novelty lies in the application of fuzzy rules on network metrics. We also provide modifiable fuzzy rules to expand or contract the conceptual bases as required.

We organize the paper as follows. We discuss the method in Section 2. We illustrate the outcomes of the approach using an example of RNA virus antivirals in Section 3. We also review the outcomes and limitations in that section, followed by

Proceedings of the 2nd Workshop on Data Science with Human in the Loop: Language Advances, pages 1–7
June 11, 2021. ©2021 Association for Computational Linguistics

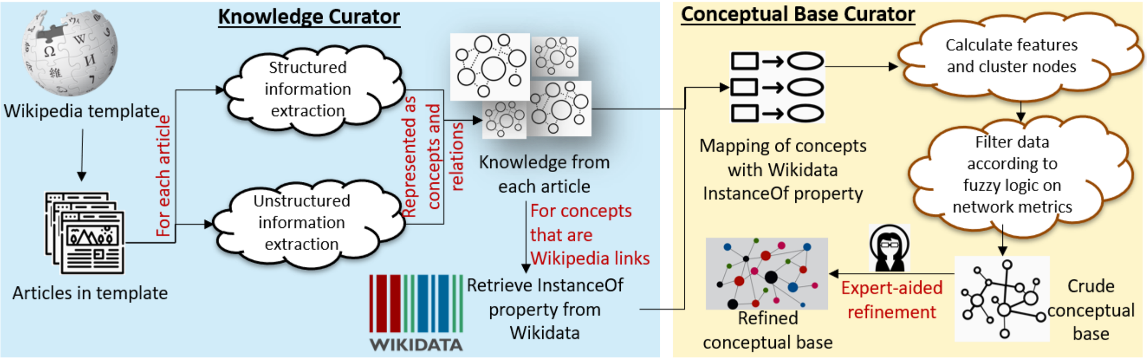
Knowledge Curator
Wikipedia template
Articles in template
For each article
Structured information extraction
Unstructured information extraction
Represented as concepts and relations
Knowledge from each article
For concepts that are Wikipedia links
Retrieve InstanceOf property from Wikidata
WIKIDATA
Conceptual Base Curator
Mapping of concepts with Wikidata InstanceOf property
Calculate features and cluster nodes
Filter data according to fuzzy logic on network metrics
Crude conceptual base
Expert-aided refinement
Refined conceptual base

Figure 1: Method overview

related works in Section 4. We conclude the paper in Section 5.

2 Proposed Method

We show the overview of the method in Figure 1. Our system consists of two parts: Knowledge Curator and Conceptual Base Curator. The Knowledge Curator extracts information from articles and Wikidata to construct a basic knowledge graph, and the Conceptual Base Curator employs machine learning techniques for processing and fuzzy rules to filter the relevant concepts.

2.1 Knowledge Curator

Collecting articles Our framework uses the template name as an input to gather information from a particular domain. The pattern "Template:Template name>" defines the Wikipedia templates. We use Wikipedia's special export webpage[1] to export the template's data into XML for faster processing. To remove unnecessary text from the XML, we use pattern-based cleaning and rule-based parsing. To retrieve the article names in the template, we use rule-based parsing. We export the information as XML for each article and use pattern-based cleaning to clean it.

Information extraction from articles Structured material, such as content information and infoboxes, can be found in Wikipedia articles. In the same way, they contain unstructured information in the context of the article's text. We extract this information by rule-based text processing on the cleaned article's XMLs.

Graph representation We represent the extracted information as a graph for further processing. For each article, we create a separate knowledge graph where the article node is the central node. We add section-subsection information using the relations: has_section and has_subsection. We add infobox information by adding has_ in front of the first column labels of the infobox and the first column label as the node. For example, Earth[2] info box contains information on mass. We add the has_mass relation to the *Earth* node with *mass* node. We process the text in the article by text normalization and sentence segmentation[3]. We tokenize[4] the sentences and extract noun chunks[5] from the sentences and consider the noun chunks as the nodes. We join the first noun chunk of the sentences with the section node using the relation has_info_about. The trailing noun chunks are added to the previous noun chunk nodes using in-between tokens as the relation. We also create a list of nodes that are links to other articles.

Retrieving Wikidata instance For all the nodes that are links to other Wikipedia articles, we parse the instanceOf[6] property using web crawling and save them to a file.

2.2 Conceptual Base Curator

Mapping We map the nodes to their Wikidata instances. If an instance is present, we replace the node with the instance name. If there are multiple instances, we create multiple nodes and add all the connecting nodes to the instance nodes. For example, a node A is connected to node B and C and A has Wikidata instanceOf as A_{i1} and A_{i2}. Then we

[1] https://en.wikipedia.org/wiki/Special:Export

[2] https://en.wikipedia.org/wiki/Earth

[3] https://spacy.io/usage/linguistic-features#sbd

[4] https://spacy.io/usage/linguistic-features#tokenization

[5] https://spacy.io/usage/linguistic-features#dependency-parse

[6] https://www.wikidata.org/wiki/Property:P31

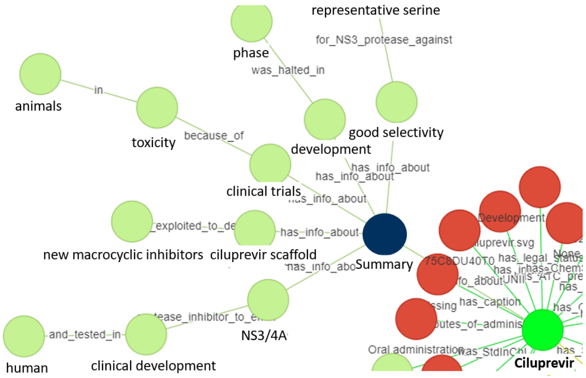

Figure 2: Screenshot of a small part of graph for *Ciluprevir* drug: the dark green node is the article node. Red, navy blue and light green nodes are information from infoboxes, section and text, respectively. Light green nodes constitutes noun chunk information connected via in-between tokens or has_info_about relations.

replace A-B and A-C with A_{i1}-B, A_{i1}-C, A_{i2}-B, A_{i2}-C in the graph.

Clustering nodes There are several similar nodes in the graphs of all the articles. We calculate the Levenshtein distance (Levenshtein, 1966) based feature matrix for all the nodes. We perform affinity propagation clustering (Frey and Dueck, 2007) which outputs cluster and cluster exemplars. We replace the nodes in the clusters with their exemplars for further use.

Node filtering and knowledge graphs collation The uncertainty factor of the concepts is the impetus for using fuzzy logic to construct a conceptual base. If we fill the conceptual base with all possible notions, the structure assumes that all concepts and relations are equally representative of the domain. However, this is not the case. Some notions are more applicable than others. Consider the following three medications: Remdesivir[7], Ledipasvir[8] and Dasabuvir[9]. Medical uses, side effects, and trade names are all common concepts. As a result, these can be said to be true in the drug domain with some certainty. The Remdesivir article contains information about medical usage controversy, which is absent in other drugs. As a consequence, this concept can be categorized as less significant.

We use fuzzy logic to find relevant concepts in a particular domain. For this, we filter out the nodes whose relation does not contain a word with VERB pos-tag. We collate the graphs for all the articles and remove "a, an, the" from the nodes.

Fuzzy logic on network metrics We calculate two network metrics: degree centrality and betweenness centrality (Freeman, 1977). The centrality metric identifies the network's most influential nodes. The number of connections a node has determines its degree centrality. The degree centrality of a vertex v, for a given graph $G := (V, E)$ with $|V|$ vertices and $|E|$ edges, is defined as:

$$C_{Deg}(v) = deg(v) \tag{1}$$

Where, $deg(v)$ is the degree of vertex v. The number of times a node appears in the shortest path of other nodes is known as betweenness centrality. It is a metric that reflects a node's power over other network nodes. It is defined by the equation:

$$C_{Btw}(v) = \sum_{i \neq v \neq j} \frac{\sigma_{ij}(v)}{\sigma_{ij}} \tag{2}$$

where σ_{ij} is the total number of shortest paths from node i to node j and $\sigma_{ij}(v)$ is the number of those paths that pass through v.

The fuzzy logic uses the above-defined network metrics to decide the relevancy of the concepts. The fuzzy logic consists of four main components: fuzzifier, rule base, inference engine, and defuzzifier. Fuzzifier converts inputs to fuzzy sets characterized by membership functions (MF). Rule base consists of IF-THEN rules used to drive the inference engine. The inference engine makes fuzzy inference on the fuzzy input based on the defined rules. Defuzzifier converts fuzzy set to the required output.

In our system, the input is degree centrality and betweenness centrality measures for all the nodes. We have experimented with the Gaussian membership function. The Gaussian MF is defined as:

$$GaussMF(x; \mu, \sigma) = e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{3}$$

where, x is the input, μ is the mean and σ is the standard deviation of x. We generate gaussian MF for both the centrality measures.

We use categorical inference on the concept relevance (HIGH, MEDIUM, LOW) and Mamdani Implication for getting the output. Assuming a rule $R_i = (D_i \text{ OR } B_i) \rightarrow N_i$, is defined by $\mu_{R_i} = \mu_{D_i ORB_i \rightarrow N_j}(d, b; n)$, where μ is membership function, D_i and B_i are fuzzy sets for degree and betweenness centrality and $N_j where j \in$

[7] https://en.wikipedia.org/wiki/Remdesivir

[8] https://en.wikipedia.org/wiki/Ledipasvir/sofosbuvir

[9] https://en.wikipedia.org/wiki/Dasabuvir

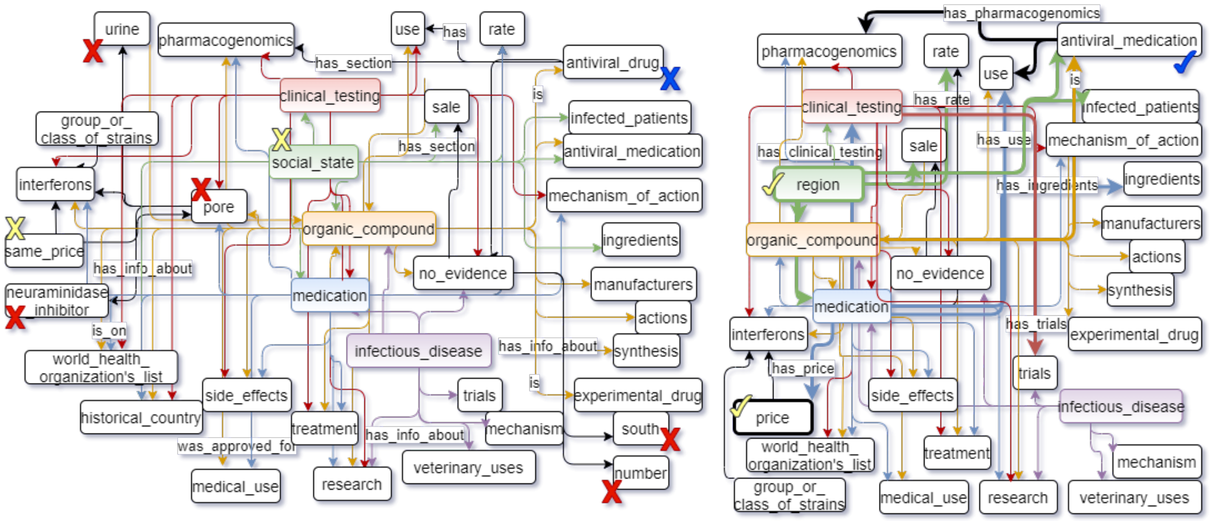

Figure 3: (a) Left: automatically generated crude conceptual base, (b) Right: refined conceptual base, consisting of concepts from section and text. (a) Left: red crosses depict nodes removed, yellow crosses depict the modified nodes, and blue crosses depict the node merged to another node. (b) Right: refined nodes and edges are shown in bold. Yellow ticked nodes are modified, and blue ticked are merged. For clarity, we show the five most central nodes and nodes connecting to them with different colors.

$[1, 2, 3] \equiv [HIGH, MEDIUM, LOW]$ denotes relevance set for nodes. Then, the Mamdani Implication uses minimum operator ($\wedge$) for fuzzy implication.

$$\begin{aligned} \mu_{N_j}(n) &= \alpha_i \wedge \mu_{N_j}(n) \\ where, \alpha_i &= (\mu_{D_i} \wedge \mu_{B_i}) \end{aligned} \tag{4}$$

We define three rules for inference:

- IF $\mu_{d_n} \wedge \mu_{b_n}$ <= 0.6 THEN $\mu_{N_j}(n) = HIGH$
- IF 0.6 < $\mu_{d_n} \wedge \mu_{b_n}$ <= 0.8 THEN $\mu_{N_j}(n) = MEDIUM$
- IF $\mu_{d_n} \wedge \mu_{b_n}$ > 0.8 THEN $\mu_{N_j}(n) = LOW$

The values in the rules are modifiable to increase or decrease the span of concepts covered in various relevance levels.

We filter out node-edge-node pairs using nodes of varying significance. We consider a node-edge-node pair highly relevant if any node in the pair is highly relevant and the node-edge-node pair has appeared in more than two articles. Similarly, we translate the medium and low importance at node level to node-edge-node pair level. We only use highly relevant node-edge-node pairs in this paper, but medium and low relevance pairs may be added to extend the conceptual base if required. We measure the resultant network's largest connected component and present it to the domain expert for further refinement.

Refining the concept base The domain expert refines the crude conceptual base. Removal or modification of semantically related concepts and removal or modification of notions that reflect the same object are both parts of the refinement process. The expert makes node connections to the modified nodes by naming "has_<node>" to new relations. There is no modification of the relations where the node is not modified.

3 Results and Discussion

3.1 Case Study on RNA Virus Antiviral Drugs

We present the results of our approach using an example of RNA virus antiviral drugs[10]. The system is implemented in Python. All the steps automatically retrieve or process the data until stated otherwise.

The system first curates the knowledge graph from all the articles using the section, infobox, and text information. We show a screenshot of a small part of the knowledge graph for the *Ciluprevir* drug in Figure 2. The system also retrieves and maps the Wikidata instanceOf property to all the link-based nodes.

Next, we apply affinity propagation to cluster similar information together. We experimented by clustering section, infoboxes and text together and independently. Infoboxes present a structured summary of the article's information. We note that the

[10] https://en.wikipedia.org/wiki/Template:RNA_antivirals

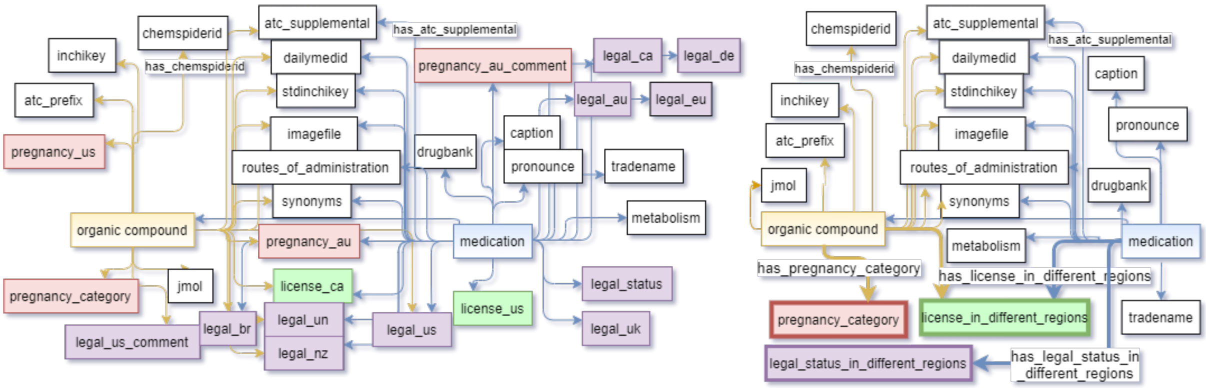

Figure 4: (a) Left: Crude and (b) Right: refined conceptual base from infoboxes. Same color nodes represent instances of the same concept.

clustering of infoboxes tends to lose information because different information identifiers may come under a single cluster, although they represent independent information. As a result, the final conceptual base contains minimal information from infoboxes. In our experiment, the automatically created conceptual base that uses clusters of all information together contains 49.4%, 4%, and 46.6% concepts coming from section, infoboxes, and text, respectively. Hence, we cluster only section and text information (independently) and use infoboxes information as it is.

After this, the application of fuzzy logic results in a crude conceptual base. Due to space restrictions, we show snippets of the model with only a few mentions of the edge names: consisting of section and text information in Figure 3(a) and infobox information in Figure 4(a). The respective refined models are shown in Figure 3(b) and 4(b). Edges or relations mostly consists of names such as has_info_about, has_section, has_subsection, has_type and verbs such as is, approved_by, is_not_recommended_during, etc. We call our output conceptual base and not conceptual model because the relations such as has_info_about, has_section, has_subsection does not provide any meaningful link between the concepts. Meaningful modification of such relations can be considered as a downstream task.

The templates contain few articles of different domains as well. For instance, RNA antiviral template contains disease and virus names as well. But, the proposed approach ensures that we consider only statistically significant concepts for the conceptual base. We manually validate that the crude conceptual base contains 14%, 16%, and 70% of concepts from section, text, and infoboxes, respectively.

3.2 Discussion

Our observations suggest that the crude conceptual base can capture most of the relevant information from both the section and text information and infobox information. There are few ambiguous names in the nodes like *pore*, *south*, *rate* (marked using crosses) in Figure 3(a) and *legal_us*, *legal_uk*, etc. (colored nodes) in Figure 4(a), which the domain expert removes or corrects. The expert also modifies edges, where nodes are modified.

The crude base contains two types of nodes: 1) nodes representing the same object in the current context but can have different meanings, and 2) nodes that are instances of another concept. For example, in Figure 3(a), the nodes *medication*, *antiviral drug* and *antiviral medication* represents antiviral drugs in the current context. These nodes appear because an article node is the most central in their knowledge graph, and they can have multiple Wikipedia instanceOf properties. Similarly, there are many instances of legal status and pregnancy category in Figure 4(a). Instances appear in the infobox conceptual base because we do not cluster those nodes. As a result, original data is retained for calculation of relevance.

Since the refinement process is manual, the expert can decide how to modify the crude conceptual base as per the need. In Figure 3(b) and 4(b), we have shown basic refinement. In Figure 3(a), we show red crosses on the nodes that are removed because of ambiguity or no meaningful information, yellow crosses on the nodes that are modified because of inappropriate names but are meaningful, and blue cross on the node that is merged with another similar node. Here, *antiviral drug* is merged to *antiviral medication*. The refined version in Figure 3(b) depicts bold boundary nodes and edges that the expert modifies. In Figure 4(a), same-colored nodes represent instances of same concepts, which the expert merge into one in Figure 4(b).

In the presented case study, the expert modifies about 30% of the total nodes (section+ text+ infoboxes). However, this is subject to the structure of Wikipedia articles in the navigational template. For example, most of the articles in the Distillation[11] template do not contain infoboxes, which reduces the percentage of nodes that needs to be modified.

Following are the limitations of our approach:

- Parsing information from web pages is a time-consuming task, so we use XML and text processing for information gathering. Sometimes, rule-based text processing incorrectly extracts the information, and seldom, the XML does not contain full information. We manually check the infobox content after cleaning the XMLs. We find that approximately 47.5% of infobox entries are incorrect or empty in our case study. In the future, we plan to check the performance and scalability of other tools.
- Sections constitute a small part of the article's information, but we lose a considerable amount of textual information because of the filtering process. We are currently exploring techniques to create enhanced knowledge graphs using language models where filtration of nodes results in minimum or no information loss.
- Currently, we do not provide any aid for refining the conceptual base. We plan to create a GUI for this purpose that will include controllers for fuzzy logic and an interface for effortless refinement, further reducing the time and effort needed to create the conceptual base.

4 Related Works

Many researchers have worked on fuzzy ontology creation and their downstream applications, such as generating taxonomies, ontologies, and conceptual models from various data sources.

The use of fuzzy logic for creating concept lattices and ontologies has been studied previously by various researchers. There have been studies regarding fuzzy ontology creation (De Maio et al., 2009), (Tho et al., 2006), using fuzzy ontology and concept models in various domain-specific tasks and dataset (Parry, 2006), (Abulaish, 2009), (Quach and Hoang, 2018). As opposed to the previous work, we employ fuzzy logic using network metrics attributes.

There are a few significant open-domain, community-driven projects for structured knowledge creation. DBpedia (Lehmann et al., 2015) extracts structured information in multiple languages from Wikipedia infoboxes. Yago (Suchanek et al., 2007) has released various versions, and this also uses Wikipedia infoboxes. It also employs Wikipedia categories to determine the type of information, which is then mapped to WordNet taxonomy. Wikidata (Vrandečić and Krötzsch, 2014) a collaborative database, also links Wikipedia data with unique identifiers. Apart from the community-driven projects, researchers also used Wikipedia in other open-domain tasks such as document topic classification (Hassan et al., 2012), collaborative ontology creation (Hepp et al., 2006), semantic conceptual modeling and semantic relatedness interpretation (Saif et al., 2018), explaining facts in AI (Sarker et al., 2020), learning named entities (Nothman et al., 2013), large-scale taxonomy generation (Ponzetto and Strube, 2007). Researchers also used Wikipedia in domain-specific tasks like exploiting Wikipedia knowledge for classification tasks (Warren, 2012) and extracting domain-terms and terminologies from Wikipedia (Vivaldi and Rodríguez, 2010), (Vivaldi and Rodríguez, 2011), (Vivaldi et al., 2012). In this research, we provide domain-specific conceptual base construction from a small set of articles extracted on-the-fly from Wikipedia navigational templates instead of full Wiki dumps or other domain-specific corpora/texts. We also exploit unstructured text in addition to the structured information like Wikipedia info-boxes and article content structure.

5 Conclusion

We use Wikipedia navigational templates to build domain-specific conceptual bases in this study. To compute the relevance of the concepts, our system generates a graph representation of the article's knowledge and uses fuzzy logic on top of its network metrics. With a bit of human intervention, the system outputs a refined conceptual base that can be used further for various downstream purposes.

[11] https://en.wikipedia.org/wiki/Template:Distillation

References

Muhammad Abulaish. 2009. An ontology enhancement framework to accommodate imprecise concepts and relations. *Journal of Emerging technologies in web intelligence*, 1(1).

Carmen De Maio, Giuseppe Fenza, Vincenzo Loia, and Sabrina Senatore. 2009. Towards an automatic fuzzy ontology generation. In *2009 IEEE International Conference on Fuzzy Systems*, pages 1044–1049. IEEE.

Frederico Fonseca and James Martin. 2007. Learning the differences between ontologies and conceptual schemas through ontology-driven information systems. *Journal of the Association for Information Systems*, 8(2):4.

Linton C Freeman. 1977. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41.

Brendan J Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *science*, 315(5814):972–976.

Mostafa M Hassan, Fakhri Karray, and Mohamed S Kamel. 2012. Automatic document topic identification using wikipedia hierarchical ontology. In *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, pages 237–242. IEEE.

Martin Hepp, Daniel Bachlechner, and Katharina Siorpaes. 2006. Harvesting wiki consensus-using wikipedia entries as ontology elements. In *SemWiki*. Citeseer.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R Curran. 2013. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151–175.

David Parry. 2006. Fuzzy ontologies for information retrieval on the www. In *Capturing Intelligence*, volume 1, pages 21–48. Elsevier.

Simone Paolo Ponzetto and Michael Strube. 2007. Deriving a large scale taxonomy from wikipedia. In *AAAI*, volume 7, pages 1440–1445.

Xuan Hung Quach and Thi Lan Giao Hoang. 2018. Fuzzy ontology modeling by utilizing fuzzy set and fuzzy description logic. In *Modern Approaches for Intelligent Information and Database Systems*, pages 15–26. Springer.

Abdulgabbar Saif, Nazlia Omar, Mohd Juzaiddin Ab Aziz, Ummi Zakiah Zainodin, and Naomie Salim. 2018. Semantic concept model using wikipedia semantic features. *Journal of Information Science*, 44(4):526–551.

Md Kamruzzaman Sarker, Joshua Schwartz, Pascal Hitzler, Lu Zhou, Srikanth Nadella, Brandon Minnery, Ion Juvina, Michael L Raymer, and William R Aue. 2020. Wikipedia knowledge graph for explainable ai. In *Iberoamerican Knowledge Graphs and Semantic Web Conference*, pages 72–87. Springer.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Quan Thanh Tho, Siu Cheung Hui, Alvis Cheuk M Fong, and Tru Hoang Cao. 2006. Automatic fuzzy ontology generation for semantic web. *IEEE transactions on knowledge and data engineering*, 18(6):842–856.

Jorge Vivaldi, Luis Adrián Cabrera-Diego, Gerardo Sierra, and María Pozzi. 2012. Using wikipedia to validate the terminology found in a corpus of basic textbooks. In *LREC*, pages 3820–3827.

Jorge Vivaldi and Horacio Rodríguez. 2010. Finding domain terms using wikipedia. In *LREC*.

Jorge Vivaldi and Horacio Rodríguez. 2011. Extracting terminology from wikipedia. *Procesamiento del lenguaje natural*, 47:65–73.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Robert Warren. 2012. Creating specialized ontologies using wikipedia: The muninn experience. *Berlin, DE: Proceedings of Wikipedia Academy: Research and Free Knowledge (WPAC2012). URL: http://hangingtogether. org*.

Wikipedia. 2021a. Wikipedia category contents. `https://en.wikipedia.org/wiki/Wikipedia:Contents/Categories`.

Wikipedia. 2021b. Wikipedia navigation template. `https://en.wikipedia.org/wiki/Wikipedia:Navigation_template`.