Towards Semi-Automatic Approach of Building an Ontology: A Case Study on Material Handling Data

Sepideh Sadat Sobhgol[1] [a], Mario Thron[1] [b] and Giuliano Persico[2] [c]
[1] *ifak e.V. Magdeburg, Germany*
[2] *Demag Cranes & Components GmbH, Germany*

Keywords: Ontology, Semantic Search, Keyword Extraction, Term Relation, Similarity Measurements.

Abstract: InnoSale project aims to improve sales processes for complex industrial equipment and services using AI technologies. The project addresses the challenges of time-consuming back-office support and interpreting customer requests using different vocabularies. As partners involved in the project, we are developing a semi-automated approach to the creation of an ontology for the material handling domain by merging existing terminology from leading companies in the industry. This ontology will serve as the basis for a semantic search engine to improve the generation of quotations and the matching of customer requirements. Through the use of historical data and advanced machine learning techniques, the search engine streamlines the sales process, reducing manual effort and improving response times. The results showcases how the utilization of machine learning and NLP techniques can aid in constructing an ontology in a semi-automatic fashion. The study demonstrates the effectiveness of extracting terms, identifying synonyms, and uncovering various relationships, contributing to the development of an ontology. These approaches offer potential for improving the ontology construction process and enhancing semantic search capabilities, leading to more effective information retrieval. This position paper, being concise in nature, presents our initial findings and progress in this endeavor. It's important to note that, based on new sources of information and ongoing research in the future, the results and conclusions may evolve or differ.

1 INTRODUCTION

The production of industrial goods that involves creating a variety of products by adding multiple options to base products is often referred to as modular production or modular manufacturing. It offers two approaches to product customisation: pre-existing product options sourced from a catalogue, or custom options created for a unique case. Off-the-shelf options allow for quick and efficient customisation, while custom options offer the opportunity to create a truly unique product tailored to a specific customer's needs, but require more time and resources to develop and test.

Finding similar custom options in a manufacturer's project history can be an efficient approach to creating new custom options. However, finding these similarities in a project history spanning several decades can be challenging and time consuming. One of the goals of the InnoSale project was to establish efficient methods for identifying and exploiting pre-existing custom options that are related to a new customer request.

Our semantic search approach involves roughly the following steps:

- Generate ontology (semi-automatically)
- Map custom option projects to ontology concepts
- Search those projects based on concept mappings

This article describes our advances with regard to semi-automatically generating an ontology for a modular production domain, especially for the material transportation domain. An example of such a custom option is a heat shield for bearings or gears of a transportation system when it is to be installed in a steel plant and the heat is coming from a direction specific to that customer.

In project documents, the technical terms commonly used by experts can differ significantly from the layman's terms used in customer requests. This

[a] https://orcid.org/0000-0002-9746-3612
[b] https://orcid.org/0009-0002-0648-9903
[c] https://orcid.org/0009-0008-5050-473X

Sobhgol, S., Thron, M. and Persico, G.
Towards Semi-Automatic Approach of Building an Ontology: A Case Study on Material Handling Data.
DOI: 10.5220/0012210000003598
In *Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2023) - Volume 2: KEOD*, pages 248-254
ISBN: 978-989-758-671-2; ISSN: 2184-3228

disparity necessitates the use of an ontology that establishes synonym relationships between these terms. By incorporating an ontology, semantic search techniques can yield more accurate and relevant results than a simplistic word indexing approach.

The discipline of ontology is introduced by philosophers such as Aristotle, but has been defined differently for use in computer science by several authors. A summary is given in (Guarino et al., 2009). The authors distinguish between the original definition of an ontology in computer science as an ”explicit specification of a conceptualization” (Gruber, 1995) and an extended definition by later publications as a ”formal specification of a shared conceptualization” (Borst, 1997). For our purposes, the term ”shared” is not important, since the terms used in project documents and by customers of the modular product manufacturer are usually non-shareable intellectual property of that company. The term ”formal” implies the use of a formal exchange format for ontologies such as OWL, the Web Ontology Language (Baader et al., 2005; Motik et al., 2012), which is not required for our non-shared ontology. The usability of our semantic search approach depends a lot on its performance and thus, we focus more on efficient custom data structures than on exchange of the ontology. Therefore, we use the term ”ontology” in its original, more general sense, as defined in (Gruber, 1995) and not in its more modern meaning.

There is no automated process for ontology modeling. Terms, such as nouns or important/frequent terms in the domain, are used to create the concept space. However, manual effort from domain experts is still necessary to form concepts and relationships in the ontology. To reduce this manual effort, machine learning (ML) and natural language processing (NLP) can be utilized, especially when Description Logic (DL)-based languages are not used. Different layers in ontology learning and building are shown in Fig 1. The upper layers are based on terminology and terms with similar meanings, represented in the lower layers. Generating terms and their synonyms semi-automatically can help reduce the workload for domain experts.

In recent years, semantic search engines have gained attention due to their ability to provide more comprehensive and accurate search results compared to traditional word index-based search engines. There are several approaches to semantic search, which are based e.g. on word embeddings or on ontologies. In InnoSale project, we explore an ontology-based semantic search approach that can retrieve results when synonyms or generalisations of the search terms are present in the target documents. In this article, we

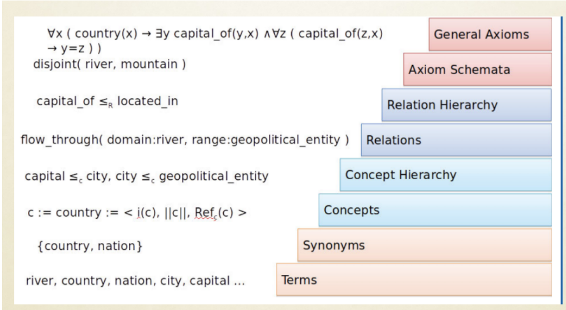

Figure 1: Ontology Learning Layer Cake (Cimiano et al., 2009).

discuss an efficient method for creating an ontology from existing sources, such as controlled vocabularies, that can be used for semantic search, which then could reduce the gap between searching the specific words in documents and keywords in query (Ramkumar and Poorna, 2014).

At first, our ontology will be a data structure which unifies the terminologies of project partners in material handling domain by bringing synonyms together to create a concept or finding a concept hierarchy. Further-on, the unified ontology also needs to be updated by new terms, which are extracted from incoming inquiries and which cannot be found in the original version of the unified ontology. Therefore, Named Entity Recognition (NER) (Al-Moslmi et al., 2020) can be used to extract the underlying keywords from customer inquiries in terms of analyzing text at the word or subword level. Discovering the entities unveils the underlying structure of the data and thus better serves as a step towards semi-automatic approach of creating an ontology. For manual editing the ontology, an Ontology Editor will be developed which provides a graphical user interface for this task.

2 BACKGROUND

In this section, we briefly establish key terms and context to understand our study. We start with some explanations of ontology-based semantic search and basic listing on different methods of synonym detection.

Semantic search is a document retrieval process that goes beyond simply relying on word occurrences in documents. Instead, it leverages domain knowledge, which can be represented through an ontology—a formal specification of concepts and their relationships (Hotho et al., 2006). Ontology-based semantic search uses ontologies to understand the meaning of user queries and the content being searched. This enhances the accuracy and relevance of search results. In this approach, the search engine analyzes both the query and the data against the ontology, enabling it to interpret the query's intent and uncover semantic relationships between concepts. This understanding of semantics allows the search engine to provide more precise and contextually relevant search results. Ontologies in semantic search provide more advanced capabilities than keyword-based search. They consider related concepts, synonyms, hierarchical relationships, and other semantic connections to deliver more accurate and comprehensive results. Ontology-based semantic search is particularly advantageous in domains with intricate and specialized terminology, where comprehending the semantic context plays a crucial role in retrieving pertinent information (Ding et al., 2004; Mangold, 2007). The connection between documents and ontologies plays a crucial role in semantic search approaches. There are two main approaches: tight coupling and loose coupling. In tight coupling, documents explicitly refer to concepts in the ontology, making it easier to resolve homonymies. However, it requires significant effort in annotating documents with semantic information. On the other hand, in loose coupling, documents are not bound to a specific ontology, which presents the challenge of selecting the appropriate ontology. While loose coupling provides flexibility, it has limitations in terms of semantic resolution, especially in scenarios like the World Wide Web. Ontology-based semantic search engines utilize ontologies, comprising concepts, properties, constraints and axioms. Standard properties including synonym-of, hypernym-of, meronym-of, instance-of, negation-of are used to capture relationships in semantic search, enhancing capabilities but introducing dependencies on ontology structure (Hotho et al., 2006).

In order to enhance the ontology's richness by incorporating synonym relationships between terms, one common technique is based on linguistic resources, such as dictionaries and thesauri, which provide explicit synonyms for a given word. Another approach involves utilizing corpus-based methods, where large collections of text are analyzed to identify co-occurring words or patterns that indicate synonymy. Additionally, distributional similarity methods leverage word embeddings or vector representations to measure the semantic similarity between words and identify synonyms. Machine learning techniques, including supervised and unsupervised algorithms, have also been employed for synonym and relation extraction by training models on annotated datasets or using clustering algorithms to group similar words (Zelenko et al., 2003; Nguyen and Grishman, 2015; Han et al., 2020; Mohammed, 2020). Finally, hybrid approaches combining multiple methods have shown promise in achieving more accurate synonym detection results (Blondel and Senellart, 2002; Wang and Hirst, 2009; Yıldız et al., 2014).

3 EXISTING TERMINOLOGIES

The considered project partners already maintained and still maintain different terminologies, which define the vocabulary to be used for naming of products and parts in their projects. The terminologies shall be unified into a single terminology. It will cover a broader set of terms used by the technical experts than one of the existing terminologies. To ensure data privacy rights, we are unable to upload our terminology. However, we will describe the structure of those files here instead. Table 1 and Table 2 present an Excel file that illustrates the structure of one of the current terminologies. Fig 2 illustrates the structure of the Acrolinx Database. These terminologies include translations in various languages. The Acrolinx database may also include synonyms for terms. In Acrolinx, terms that are synonyms in different languages are assigned the same entry ID.

Table 1: Standard Terms.a.

DE	EN	ES
Abdeckband	Masking tape	Banda protectora
Abdeckblech	Cover plate	Placa protectora
Abdeckblech Geräteseite	Cover plate equipm.side	Chapa prot. lad aparell.

Table 2: Standard Terms.b.

FR	IT	CS
Bande de protection	Nastro coprente	Zakrývací páska
Tôle de protection	Lamiera copertura	Krycí plech
Tôle protect. c. appareil	Lamiera copert.apparecchi	Krycí plech strany stroje

4 STEPS TOWARDS SEMI-AUTOMATIC CREATION OF AN ONTOLOGY

This section outlines the steps involved in creating an ontology. The creation of the ontology involves the following steps:

- Importing existing terminologies: The vocabulary used by the manufacturers is defined in terminologies which is in a format of an **Excel file** and **Acrolinx database** for this purpose. These terminologies include translations in different languages, and efforts should be made to unify them.
- Creating an unified terminology: The **unified terminology** incorporates term abstractions to enable

```xml
<entry>
    <value field="entry/id">13</value>
    <value field="entry/uuid">a918fccf-4a94</value>
    <head>
        <term>
            <name>return air</name>
            <language>en</language>
            <value field="term/domain">undefined domain</value>
            <value field="term/uuid">120bdf01-4818</value>
            <value field="term/id">1371197695606</value>
        </term>
    </head>
    <term>
        <name>Abluftmenge</name>
        <language>de</language>
        <value field="term/domain">undefined domain</value>
        <value field="term/uuid">44641d47-ddf3</value>
        <value field="term/id">1371197695646</value>
    </term>
    <term>
        <name>débit d'air repris</name>
        <language>fr</language>
        <value field="term/domain">undefined domain</value>
        <value field="term/uuid">5334a674-848b-44a3/value>
        <value field="term/id">1371197695614</value>
    </term>
    <term>
        <name>débit d'évacuation</name>
        <language>fr</language>
        <value field="term/domain">undefined domain</value>
        <value field="term/uuid">38e4ce12-4175</value>
        <value field="term/id">1371197695654</value>
    </term>
</entry>
```

Figure 2: Acrolinx.

the identification of product variants, thus forming an **ontology**.

- Regular updates based on incoming inquiries: Manufacturers receive inquiry emails from customers who may use different vocabulary than what was used in previous projects. The ontology should be updated with new words, which may need to be manually linked as synonym terms or term abstractions.

The following section provides further details on this approach.

4.1 Unifying Terminologies

As mentioned in Section 1 and depicted in Fig 1, in the lower layer of the ontology cake, synonyms can be connected to form a concept. Subsequently, the concept hierarchy can be explored. Fig 3 provides us with an understanding of how we can unify the terminologies to create our new term structure. Terms and their

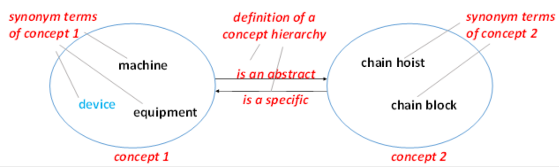

Figure 3: Terms, synonyms, concepts, abstract-specification relations.

synonyms can be stored as complex character string-based tables. However, in terms of space efficiency, it is more advantageous to store terms in a database table, with one column for the term (as a string) and the other for the termID (as an integer). This approach allows for easier establishment of relations between terms based on the integer IDs. Subsequently, if there is a relation between terms, they can be connected to each other using their respective "IDs". This can be achieved through the use of relational or NoSQL databases or by simply storing them in files. The termID can serve as a unique identifier, which may already exist in the Acrolinx term database. However, if the term is sourced from an Excel file or is entirely new, a new termID must be generated for it. Synonyms are assigned the same conceptID, and concepts can have various types of relations with each other, including abstract-specification relations. Fig 4 shows the data model of the ontology. We have defined three entities. The term entity represents a structure of how terms are being stored in database. The concept entity shows that synonyms are assigned the same conceptID and concept_concept entity are representative of the relations between concepts. The data structure is stored in Sqlite, which functions as a relational database. A Python script is implemented to build the database and its corresponding tables. Furthermore, another Python script is utilized to import terminologies into the database and merge any existing synonyms. Totally, we have 78,603 terms in term table in different languages.

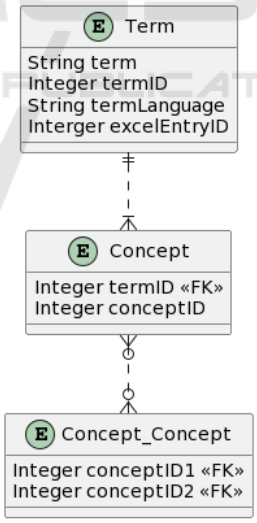

Figure 4: Ontology Data Model.

4.2 Relation Identification

In our study, we are primarily interested in two types of relations: synonym and abstract-specification relations. We examine synonym relations to address the variation in vocabulary used by customers and experts in inquiry emails. Utilizing synonyms helps identify the most similar previous projects, aiding in the current inquiry process. Additionally, the unified terminology should integrate term abstract-specification relation to facilitate the discovery of product variants as well. In following sections, we will discuss our approach to finding those relations.

4.2.1 Synonym Detection

As mentioned earlier, the Acrolinx database contains synonyms for certain terms. However, we are also interested in exploring techniques to expand the number of synonyms, near synonyms, or similar terms. Firstly, we calculated the word vector for each term using SentenceTransformer in Python. Next, we utilized a distance function provided by SciPy in Python to calculate the similarity between the word vectors using cosine similarity. We set the threshold to be greater than 0.6 in order to avoid obtaining a large number of irrelevant terms. In Table 3 and Table 4, we present similar terms to the term ’cover’ as well as family terms related to ’cover’ in both English and German. Here, we notice that when a target term and a similar term share a common part, it is more likely for the model to perceive these two terms as highly similar to each other. We have also observed that although there is a degree of semantic similarity between the target and returned terms, however, these terms cannot be considered exact synonyms and do not fall into the category of synonyms. Therefore, we classify them as similar terms for simplicity.

Table 3: Similar Terms in German.

Query	Similar Terms
abdeckung	Abstützung, Ablage
abdeckung komplett	Abtragbock komplett
abdeckung links	Gewindering links
abdeckung lüfter	Aufhängeumklammerung
abdeckung lüfter	Gitterabdeckung
abdeckung rechts	Gewindering rechts

Table 4: Similar Terms in English.

Query	Similar Terms
cover	support, shelf
cover complete	support stand complete
cover links	cover left, threaded ring left
cover fan	hanging bracket, grid cover
cover right	threaded ring right

4.2.2 Abstract-Specification Relation

We are also interested in another relationship, which involves integrating term abstractions and specifications. This integration aims to enhance the discovery of product variants as well. Hierarchical relations between terms in a text are essential for organizing concepts and understanding the semantic structure. As it is shown in Fig 5, for instance, consider the term ’cover’. It serves as an abstract term representing a general concept. Within this hierarchy, we can identify specific types of covers, such as ’cover plate’ and ’cover surface’, which can be considered as subcategories or instances of the abstract term. We implemented a tree-like algorithm to detect such relation (parent-child) between terms in a given domain. This structure is stored in a table within a SQLite database containing three columns: ’conceptID1’, ’conceptID2’ and ’relationType’. In this context, ’conceptID1’ represents the child, ’conceptID2’ represents the parent, and ’relationType’ denotes the nature of their connection. In summary, the identification and storage of hierarchical relations between terms in a text provide a foundation for organizing concepts and understanding semantic structures. At present, this process requires approximately 20 minutes to handle 15,000 English terms, which is longer than our initial expectations. Therefore, one of our ongoing tasks is to optimize this algorithm, aiming to achieve faster execution time.

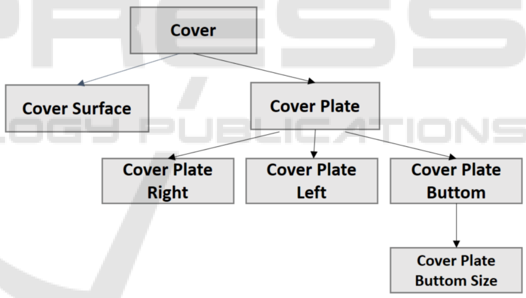

Figure 5: Hierarchical Relation between Terms.

4.2.3 Relation Identification Through Model Creation

The analytical study which is carried out in this section is mainly inspired by the work presented in (Mohammed, 2020) which addressed the problem of synonyms identification from text corpus using supervised neural network. After analyzing the previous section, it becomes evident that cosine similarity is not always indicative of synonymy. It is uncommon for the most similar word to be an actual synonym of the target word. This brings up an important question: Can we develop a system that classifies whether two words are synonyms based on their vector representations? If so, how can we acquire labeled data to train such a system. To tackle this challenge, we approached the synonymy identification problem as a classification task. As a learning algorithm, we utilized a deep learning model provided by keras library in python. The deep learning model built with Keras consists of a 128 embedded layer followed by a Dense layer, with the activation function set to softmax. With a simple model we were able to get around 86.1% accuracy on the test set. To acquire labeled training data, we retrieved pairs of synonyms from our sqlite database. These synonyms were previously populated with data from the Acrolinx database. Additionally, we included more pairs that exhibit an abstract-specification relationship, which were obtained from Section 4.2.2. The training dataset, in total, comprises 795 synonym pairs and 794 pairs demonstrating an abstract-specification relationship. For every classification run, the data was divided into training and testing sets in a 75:25 ratio, respectively. Despite the small size of the training dataset, the outcomes of the classification experiments are promising. In this binary classification scenario, the accuracy surpasses our initial expectations, indicating the viability of employing supervised learning for the task of identifying synonyms using word embeddings as features. We have reevaluated our model by testing it on pairs of terms that were not present in the training set. The results are presented in Table 5, where 0 and 1 correspond to the labels 'synonym' and 'abstract-specific' respectively. In future research, we aim to investigate whether we can minimize misclassification errors by incorporating the definition of each term as an additional parameter into this model.

Table 5: Executing Model on Test Data.

term1	term2	Actual Label	Predicted Label
kran	kran pcc-250	1	1
kettenzug	Kettenzug PKVUN	1	1
Kran	Kranbühne	1	1
Kettenumlenkrad	Kettenumlenkung	0	1
Benachrichtigung	Mitteilung	0	0
Seilscheibe Umlenkrad	Umlenkrolle	0	1
eingangswelle	welle	1	1

5 REGULAR UPDATE OF ONTOLOGY BASED ON INCOMING INQUIRIES

The manufacturer gets inquiry emails from customers, who use possibly a different vocabulary than used by experts in previous projects. The ontology should be updated accordingly by possible new terms, which need to be manually related as synonym terms or term abstractions. In order to extract keywords from incoming email, we applied Named Entity Recognition (NER) (Al-Moslmi et al., 2020) on some sample data from our project partners. We utilized a pre-trained model from the spaCy library to perform NER tasks. A named entity represents a tangible object in the real world and is assigned a label, such as 'person', 'date', 'country' and so on (Srinivasa-Desikan, 2018). To update our ontology, we have created a user interface using Angular web technology. Once the keywords are extracted, sales engineers can review and identify the relevant terms. The selected terms, along with their associated relationships, are then processed and inserted into the database. Fig 6 illustrates the various stages involved in the evaluation of a new inquiry and the subsequent update of the ontology by the ontology editor.

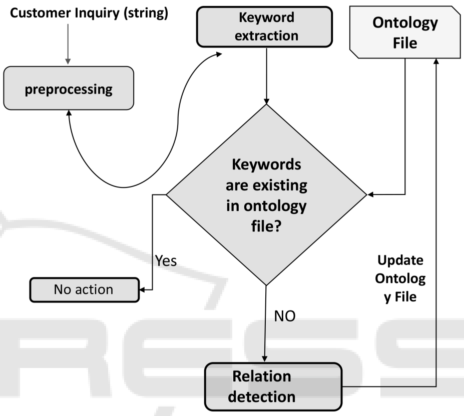

Figure 6: Evaluation of incoming customer inquiries.

6 CONCLUSIONS

This research paper presents a data-driven method for constructing an ontology. The purpose of this ontology is to facilitate semantic search for past offers or projects in relation to incoming inquiry texts. The approach employed is not entirely automatic, as a fully automatic process would result in low-quality term definitions and relationships. Our intention is not to create an ontology from scratch but to build it by utilizing existing resources, such as our terminologies. In our study, we tackled the task of detecting synonyms and abstract-specification relations using word embeddings, while also identifying hierarchical structures between terms. Sentence transformers were employed to construct word embeddings, and we conducted a qualitative evaluation of the most similar words to specific targets. Our investigation revealed that distributional similarity does not always imply synonymy; instead, similarities may be attributed to other functional factors, such as domain similarity resulting from the presence of common words between terms. Additionally, In our study, we utilized word embeddings as features to train the deep learning model. Moving forward, we aim to enhance these features by incorporating additional information inferred from the context itself, allowing the features to capture the linear contexts in which the words typically appear. This will aid in distinguishing synonymy from other sense relations. Furthermore, our current models generate a single vector representation or word embedding for each term. However, contextual models can generate word representations that are influenced by the surrounding words in the sentence (Deb and Chanda, 2022). To achieve this, it is crucial to consider the definitions of individual terms, rather than solely focusing on the terms themselves. Therefore, this paper is being presented as an ongoing project, and our next objective is to enhance the ontology by integrating term definitions sourced from additional information channels in future. Furthermore, we demonstrated that embeddings can serve as effective features for training deep learning model in classification tasks. In order to capture new terms, we utilize NER and employ an Ontology Editor to streamline the process of updating the ontology. In conclusion, this research paper illustrates that employing machine learning and NLP techniques enables the development of an ontology in a semi-automatic manner by extracting terms and detecting relations between terms. The study highlights the potential of these approaches in enhancing the ontology construction process.

ACKNOWLEDGEMENTS

This work is supported by ITEA3 under the supervision of the German Federal Ministry of Education and Research (FKZ: 01IS21084 (InnoSale)).

REFERENCES

Al-Moslmi, T., Ocaña, M. G., Opdahl, A. L., and Veres, C. (2020). Named entity extraction for knowledge graphs: A literature overview. *IEEE Access*, 8:32862–32881.

Baader, F., Horrocks, I., and Sattler, U. (2005). Description logics as ontology languages for the semantic web. In *Mechanizing mathematical reasoning*, pages 228–248. Springer.

Blondel, V. D. and Senellart, P. P. (2002). Automatic extraction of synonyms in a dictionary. *vertex*, 1:x1.

Borst, W. N. (1997). *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. Phd thesis, Universiteit Twente, Enschede.

Cimiano, P., Mädche, A., Staab, S., and Völker, J. (2009). Ontology learning. In *Handbook on ontologies*, pages 245–267. Springer.

Deb, S. and Chanda, A. K. (2022). Comparative analysis of contextual and context-free embeddings in disaster prediction from twitter data. *Machine Learning with Applications*, page 100253.

Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., Reddivari, P., Doshi, V., and Sachs, J. (2004). Swoogle: a search and metadata engine for the semantic web. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 652–659.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum. Comput. Stud.*, 43:907–928.

Guarino, N., Oberle, D., and Staab, S. (2009). *What Is an Ontology?*, pages 1–17. Springer Berlin Heidelberg, Berlin, Heidelberg.

Han, X., Gao, T., Lin, Y., Peng, H., Yang, Y., Xiao, C., Liu, Z., Li, P., Sun, M., and Zhou, J. (2020). More data, more relations, more context and more openness: A review and outlook for relation extraction. *arXiv preprint arXiv:2004.03186*.

Hotho, A., Jäschke, R., Schmitz, C., and Stumme, G. (2006). Bibsonomy: A social bookmark and publication sharing system. *International Journal*.

Mangold, C. (2007). A survey and classification of semantic search approaches. *International Journal of Metadata, Semantics and Ontologies*, 2(1):23–34.

Mohammed, N. (2020). Extracting word synonyms from text using neural approaches. *Int. Arab J. Inf. Technol.*, 17(1):45–51.

Motik, B., Patel-Schneider, P. F., Parsia, B., Bock, C., Fokoue, A., Haase, P., Hoekstra, R., Horrocks, I., Ruttenberg, A., Sattler, U., and Smith, M. (2012). OWL 2 web ontology language: Structural specification and functional-style syntax. W3c recommendation, W3C.

Nguyen, T. H. and Grishman, R. (2015). Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st workshop on vector space modeling for natural language processing*, pages 39–48.

Ramkumar, A. S. and Poorna, B. (2014). Ontology based semantic search: An introduction and a survey of current approaches. In *2014 International Conference on Intelligent Computing Applications*, pages 372–376.

Srinivasa-Desikan, B. (2018). *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras*. Packt Publishing Ltd.

Wang, T. and Hirst, G. (2009). Extracting synonyms from dictionary definitions. In *Proceedings of the International Conference RANLP-2009*, pages 471–477.

Yıldız, T., Yıldırım, S., and Diri, B. (2014). An integrated approach to automatic synonym detection in turkish corpus. In *Advances in Natural Language Processing: 9th International Conference on NLP, PolTAL 2014, Warsaw, Poland, September 17-19, 2014. Proceedings 9*, pages 116–127. Springer.

Zelenko, D., Aone, C., and Richardella, A. (2003). Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106.