Ontology modeling of the Estonian Traffic Act for self-driving buses (Extended Version)

Technical Report, Laboratory of Sociotechnical Systems, Department of Software Science, Tallinn University of Technology

Alberto Nogales [1], Ermo Täks [2] and Kuldar Taveter [2]

[1] CEIEC, Research Institute, Universidad Francisco de Vitoria (UFV), Carretera Pozuelo-Majadahonda km. 1,800, 28223 Pozuelo de Alarcón, Spain
[2] Department of Software Science, Tallinn University of Technology, Akadeemia tee 15A, 12618 Tallinn, Estonia
alberto.nogales@ceiec.es, ermo.taks@taltech.ee, kuldar.taveter@taltech.ee

Abstract. The development of self-driving cars is a major research area that has led to several still unresolved issues. One of them is the need to abide by the legal stipulations fixed by a traffic act concerning the territory of operation. Vehicles should be able to interpret traffic norms by themselves. An appropriate solution to make text understandable by machines is the use of ontologies. They provide a powerful knowledge representation mechanism. This simplifies issues that occur when, for example, a norm is modified. This paper presents a first approach where the Estonian Traffic Act is transformed from text into populated ontologies so it can be understood by machines. The proposal is a (semi)-automatic ontology learning process that combines natural language processing (NLP) and ontology matching techniques with a deep learning model. This process allows information from the raw text of the Estonian Traffic Act to be modeled with ontologies. The results show that 78% of the norms that have been considered valid can be used with the method described in the paper.

Keywords: Ontology Learning, Ontology Matching, Deep Learning.

1 Introduction

Self-driving vehicles (SDV), also known as autonomous [1], automated [2] or driverless vehicles [3], have become a technological trend in recent years. SDVs are defined in [4] as a new era of vehicle systems where a part or all of the driver's actions may be removed or limited, and where cars involve a combination of new technologies including sensors, computing power, and short-range communications, effectively creating a new human-automobile hybrid. SDVs have been part of the scientific literature for many years. The first prototype of an attempted SDV was produced in 1979 by Tsugawa et al. in Japan [5]. Their prototype had the ability to follow signaled paths. However, it was not until the end of the 2000s when they began to become a reality, with

SDVs such as those designed by Google [6]. In 2013, an experiment that consisted of a vehicle driving autonomously on an open road with traffic and no driver was accomplished [7]. The benefits of using SDVs were reviewed in [8]; it should be noted that fewer traffic accidents occur, there is a decrease in the costs and pollution, and they allow for more effective parking management.

By the end of 2016, it was announced that self-driving buses would be used in Tallinn (Estonia) starting in July 2017. These buses need to abide by the Estonian Traffic Act to move around the city. For that purpose, they must interpret norms such as the speed limit depending on the road or the meaning of traffic signals. In other words, the system responsible for driving the bus must be able to understand the text describing the traffic norms.

Legal and linguistic aspects of a legal act are tightly bound; the norm can be understood as "thought (i.e., meaning) content expressed through language" [9]. The norm, as a rule, receives expression in a norm sentence (norm formulation) and vice versa; the norm is the meaningful content of the norm formulation. Legal text has its own specific limitations in their organization and formulation because of the very function of such texts. It is usually structured in small units, articles and paragraphs; legal texts may consist only of normative statements and legal statements that must be abstract and general [10]. The smallest meaningful representation of a norm in legal text is a clause. By definition, a clause is a group of words containing a subject and predicate and functioning as a member of a complex or compound sentence.

Software run by SDVs must interpret very large and complex information. Thus, when there is a modification in the traffic act, changes are an expensive and cumbersome task. A good solution to this problem is to make a representation of legal texts that is understandable by machines. There have been several attempts to do so. Briggi et al. added metainformation to legal documents [11], [12] used logic forms based on the Prolog language, and [13] used NLP and temporal reasoning formalism. A proper way to make information understandable by machines is by using ontologies, which are defined as a specification of a conceptualization [14]; they can describe actors, relations and situations of a particular field. As a benefit, it can be remarked: discovering new knowledge by reasoning or separating the domain knowledge of the operational knowledge. By separating the operational knowledge, in the case of a norm modification, the impact of changing this information is very low [15]. In the case of the legal domain, ontologies are very useful as knowledge is updated continuously.

Therefore, by extracting the knowledge of the Estonian Traffic Act and modeling it with several ontologies, this representation of the norms produces a machine-understandable formalized representation of the norms. The process of constructing ontologies by the integration of a multitude of disciplines is referred to as ontology learning [16]. In this paper, the first approach to modeling norms from the Estonian Traffic Act as ontologies is described. The approach will consist of a hybrid process divided into four steps. Taking as a starting point the text of the Traffic Act as an XML document, it needs to be preprocessed to extract each norm as plain text. Then, the main terms from each norm will be obtained, and part of speech (PoS) tagging will be applied. This will allow the terms to be identified by their grammatical function in the sentence. In the next step, the terms will be annotated with vocabulary from a catalog called the

linked open vocabularies (LOV) [17]. This will add extra knowledge understandable by machines to the terms. Additionally, the tags obtained at the PoS stage will be annotated with an ontology that describes them. The annotation performed previously will be performed by finding mappings after using ontology matching, which is defined as the technique used to find the relationships among entities [18]. Finally, as legal text is composed of actions with different levels of restriction, a Semantic Web Rule Language (SWRL) will be applied. SWRL is a language that can be used to express rules as logical conditions; it is very useful in this work, as traffic norms are expressed as conditionals. As has been stated before, norms can be classified into different types. Thus, before applying SWRL, a deep learning neural network model can be applied to achieve this result. Deep learning is defined by [19] as a technique allowing computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction.

The rest of this paper is structured as follows: section 2 is a discussion of the state of the art of previous studies made in the fields of this paper. Section 3, "Materials and methods," gives a deeper idea of how the approach was achieved and what resources were used for that purpose. In section 4, the results are analyzed, and a use case is presented. Finally, section 5 offers some conclusions about the research and proposes future lines of work.

2 Background

There are previous studies benefitting from using ontologies to represent legal texts. [20] presents an ontology with a set of general categories and subcategories that classifies legal knowledge. Semantic Annotation for LEgal Management (SALEM) is described in [21]; this is a framework used to automatically tag Italian law using an ontology. Regarding ontology representations of traffic norms or texts, there are also several studies. An ontology for describing vehicles and traffic infrastructure in cases of complex intersections can be found in [22]. Additionally, in [23], a model based on an ontology using inference rules was offered that allows cars to reason if they must respect or relax a norm. An intelligent transportation system using ontologies is proposed in [24]; it was tested in a situation where two cars are travelling along four roads with four intersections. In [25], an ADAS ontology for autonomous driving tasks and an ADAS ontology-based map data for commercial map data was described. Finally, [26] described a conceptual description of all road entities with their interactions.

There are also several papers that use ontologies to model texts. [27] applied ontologies to documents in physics. It has also been used to expand the information of inaccessible documents by mapping their "short text" information with open sources such as DBpedia, Freebase or Yago [28]. [29] showed an approach to classify documents by using a Naïve Bayes classifier and mappings with the Yahoo! ontology for websites. [30] proposed a text tagging mechanism for document classification using the Web Ontology Language (OWL) and SWRL. Additionally, in [31], a mapping was made between a form of text called System Installation Design Principle (SIDP) and OWL.

Regarding the other techniques used in the research, deep learning has already been widely used for text classification. In [32], a convolutional neural network (CNN) was presented, introducing the use of rationales. Another approach using deep learning for text classification was used in [33], with a model called hierarchical deep learning for text classification (HDLTex). It classifies documents, both completed or fragments, depending on the hierarchy level. Another method for text classification can be found in [34]. Here, three multitask architectures of recurrent neural networks (RNN) were used to classify four text benchmarks. Additionally, in [35], a model using CNN based on the attention model was used to classify mathematic texts. Finally, in [36] CNN was used to classify information from DBpedia.

Finally, ontology learning can be found in [37], where Italian legislative texts were converted into ontologies by applying NLP, statistical text analysis and machine learning. Then, Web folksonomies were used for ontology learning in [38]. Additionally, [39] described an ontology learning system called Concept-Relation-Concept Tuple-based Ontology Learning (CRCTOL). This system uses statistical algorithms for word sense disambiguation. Another ontology learning tool called Text2Onto was applied to Spanish legal texts using language-specific algorithms [40]. Another ontology learning approach based on textual information has been proposed in [48]. Finally, in [41], ontology learning was combined with deep learning models such as CBOW and Skip-gram to a corpus of PubMed citations.

However, the present research differs from the studies cited above regarding the use of ontologies to represent text. This work describes norms from the Estonian Traffic Act using ontologies. Also related to this, the work develops an automatic process that identifies a set of different mappings between the traffic act and a set of vocabularies. It also implements a deep leaning model so norms can be automatically classified depending on their level of restriction. This classifier will simplify the task of norm categorization. Finally, ontology learning has not been previously applied in the field of texts describing the traffic laws of a country.

3 Materials and methods

As has been said before, the Estonian Traffic Act will be annotated with ontologies so machines can understand them. An ontology learning method is directly joined to the process of ontology development defined in [42]. Figure 1 shows a layer cake model of this process.

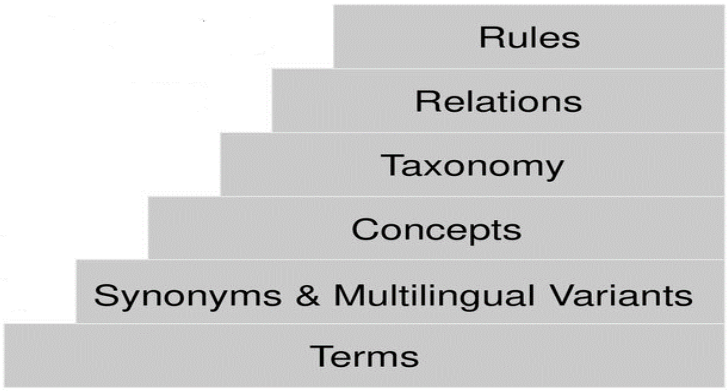

Fig. 1. Layer cake ontology learning defined by Buitelaar et al. [40]

The first step depicted in Figure 1 is related to the extraction of terms, the most basic items in a sentence. The second step consists of obtaining synonyms for these terms. In the following step, terms are related to similar concepts, even those that are from a different language. The taxonomy step involves constructing a hierarchical relation between the terms. The next step aims to discover the nonhierarchical relationships between the terms. Finally, there is a process that builds rules with all the information obtained previously.

In the following subsection, the proposed (semi)automatic process of the paper, which is divided into three stages, is explained. Section 3.1 describes the preprocessing stage of the Traffic Act, which goes from the raw XML document to the extraction of the terms for each norm. Then, PoS techniques are applied to obtain the grammatical function of these terms. These two tasks correspond to the first level of the layer cake. Section 3.2 consists of obtaining synonyms for these terms and mapping terms with ontologies, which is a similar process to the "Synonyms", "Taxonomy" and "Relations" layers. Maps of two types are obtained during this step. First, the PoS tags from section 3.1 are mapped with the OLiA Annotation Model [43]. The version for PENN Treebank PoS annotation will be used. Then, the terms plus their synonyms are mapped with LOV's vocabularies. The information from the mappings will be used to establish a relation of subclasses between the PoS tags mapping and the ones provided by the vocabularies. This is because, for example, many terms will be nouns or adjectives in a sentence. Finally, section 3.3. builds an ontology depending on the results obtained in the previous steps. This final stage has a manual task where a user decides which of the mappings fit better and an automatic task where, depending on the type of norm, the ontology is built with a different structure. To simplify the latter task, a deep learning classifier is depicted to categorize each norm depending on its level of restriction. Based on this, SWRL rules are built for the different types of norms. A picture of the process can be found in Figure 2.

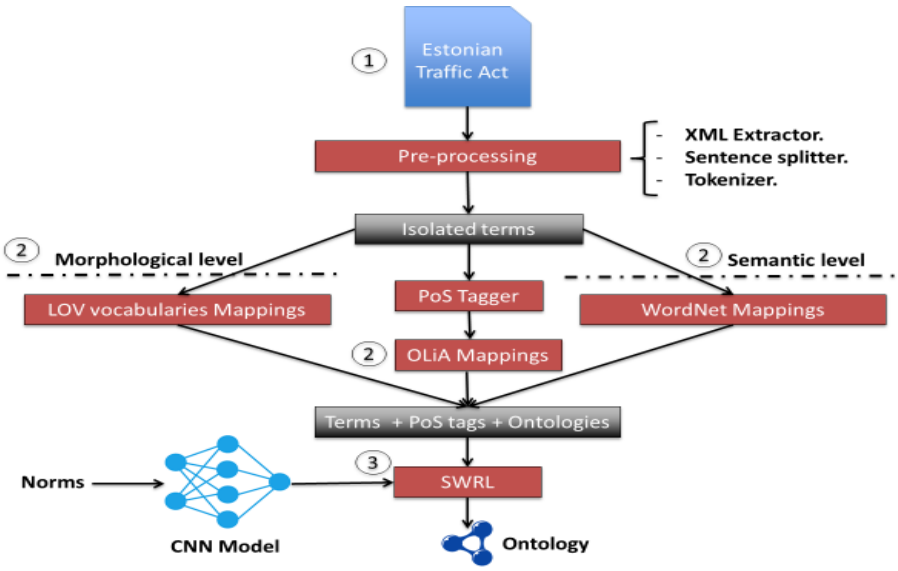

Fig. 2. Workflow of the ontology learning process.

3.1 Raw text processing

In the previous section, it was explained that the starting point is the Traffic Act of Estonia. An official version in English can be downloaded [44]; this document was translated on 07/01/2014. It is an XML document from which norms must be extracted. The problem is that some of the norms do not involve how the vehicle behaves. For example, there are norms related to the driver's levels of alcohol intoxication. Thus, all the paragraphs that do not only describe how an SDV interacts with other actors involved in traffic have been discarded.

Once these paragraphs are chosen, the process of extracting terms will start. This is typically divided into five steps: Sentence Splitter, Tokenizer, Morphological analyzer, PoS Tagger and Dependency parser. Steps one and two will be applied to that part of the process to obtain only the words that compose each norm without stop words.

Paragraphs have a set of sentences; each sentence will be considered a norm. Thus, the interest of this research lies in modeling each norm with a set of ontologies. This process starts with a "Sentence Splitter", which consists of dividing a paragraph into its sentences to process each one separately. Then, each sentence is divided into a set of words by removing its stop words; for that purpose, a "Tokenizer" is used. Both the "Sentence Splitter" and the "Tokenizer" have been developed in a Python script with the help of the NLTK package version 3.3, which is used to work with human language data. At the end of this step, a list of the words without stop words will be obtained, and the order of the words will be the same as in the sentence corresponding each list to a norm.

At this step, there is a need to know how the terms are related to understand the whole sentence. According to [45], a rule or norm in a legal text has the structure of a

case, condition, sub-condition, legal subject and legal action. This means that a subject will perform an action when a condition or some conditions are achieved. PoS tagging is the process of giving a grammatical category to every word in a sentence. This information will show which words are part of the subject, which are part of the action and which are part of the conditions.

To obtain this information, another script has been developed using a tool called a Stanford Parser [46] that is included in the NLTK package. By using the parser with a sentence, each word of a sentence will be related with a tag that defines its function in it. Tags correspond to the Penn Treebank tag set [47], which at the time of its publication contained 4.5 million words in American English. By knowing which words correspond to the conditions, which to the subject and which to the action, the norm can be built following the structure shown below.

$$if\ \{Condition==True\}\ then\ (Subject \rightarrow Action)$$

3.2 Mapping words with vocabularies

At this step, there is a need to give extra knowledge to each term extracted in the previous section. To achieve this, mappings between them and the vocabularies are needed. The mappings are performed with two different aims: the first set of mappings will consider the tags given by the PoS techniques and will map them with the OLiA annotation model which is an ontology describing PoS and the syntactic tags of Penn Treebank. The PoS mappings are easy to make as all the tags are described in the ontology.

The second set of mappings relates to the terms with all the vocabularies in LOV which is the largest catalog of vocabularies in the Semantic Web. It was decided to use this set of vocabularies, as not all the terms in the Traffic Act are covered by legal or traffic ontologies; for example, 'light'. Here, the mappings are made on two levels. The morphological level, which considers that two words are the same if they are written in the same way, and the semantic level, which considers that two words are the same if they have the same meaning. At the time of the experimentation, the catalog was composed of 601 vocabularies in different fields.

As previously mentioned, these mappings are made in two steps. The first corresponds to the morphological level and will take the set of words of a sentence obtained in the previous section and try to map it with all the vocabularies in LOV. To obtain a mapping, a word needs to be the same as a class or property of a vocabulary. To find the mappings, a Python script has been developed using the RDFLib package, which allows users to work with the resource description framework (RDF) representations. The process will consist of taking a word and comparing it to all the classes and properties of a vocabulary and then going through all the vocabularies in LOV. For the second type of mappings, there is a need to work with synonyms. In this case, two words are considered the same depending on their meaning. For that purpose, WordNet has been used, which is a large database of nouns, verbs, adjectives and adverbs grouped as cognitive synonyms [48]. Again, a Python script has been developed using RDFLib as in the previous mapping approach. To find the mappings, the synonyms of a word will be

obtained from WordNet, and then these synonyms will be compared with the terms provided by the LOV vocabularies. A mapping will be found if a synonym and a term are equal by comparing them string by string. In Table 1, there are examples of the three types of mappings described above.

Table 1. Examples of different mappings sentences.

Word	Mapped with	Type of mapping
(NN driver)	http://purl.org/olia/penn.owl#NN	PoS
Light	https://w3id.org/saref#Light	Morphological
Cycle	http://linkedgeodata.org/ontology/Motorcycle	Syntactic

3.3 Building the ontologies

The final step consists of constructing an ontology for each norm, which is a total of 420 at this point. It uses the vocabularies extracted with the vocabulary mappings to give some knowledge to the 8 words and the information given by the PoS tagging mappings. The purpose of this process is to build some logic rules that could be interpreted by machines.

First, there is the information obtained in the step of mapping the text with the vocabularies. As more than one mapping can be obtained for a word, there is a need to choose one vocabulary to represent the word. It has been decided to do this manually, which will consist of choosing the best vocabulary that fits with the meaning of this word in the sentence.

Once each word of the sentence is related to a vocabulary, the rule needs to be built based on the PoS tagging and applying SWRL. Considering [49], it is known that norms in traffic can be classified into permissions, obligations and prohibitions, each having a different representation. By using this classification of norms, it can be determined how restrictive they are. A permission denotes that the action could be done or not, an obligation denotes that the action must be done and the prohibition that the action cannot be done. Table 2 summarizes how each kind of norm can be structured.

Table 2. Norm structures.

Type of norm	Structure
Permission	if (Condition==True) then (Subject → (Action OR NOT Action))
Obligation	if (Condition==True) then (Subject → Action)
Prohibition	if (Condition==True) then (Subject → NOT Action)

To make the classification of different norms easier and more precise, a deep learning classifier has been developed. Based on [50], a convolutional neural network (CNN) was used. In this case, the first layer is an embedding one that receives as input a representation of a set of categorized norms that form the training data. Then, a structure

of convolutional and max-pooling layers was added. A total of five of these architectures were added, using kernels of sizes 3, 4, 5, 10, 30 and 50, with a total number of 128 filters. Finally, the features obtained by the convolutional stages were classified with two fully connected layers as output. Once the architecture was defined, the model was trained with 90% of the data in 6 epochs with a batch sizes of 50. The rest of the corpus was used for validation. After the training stage, the model had a loss of 9% and an accuracy of 96%. These metrics at validation time were 5% and 99%. The classifier was built with the Keras library, an API written in Python for managing neural network models

At this point, there was a set of words with their corresponding tags that denoted the grammatical function of each one. By using these tags and the type of norm depending on the category, SWRL can be applied so that machines can understand them.

4 Analysis and discussion

This section presents an analysis of the different results obtained during the research. The research used a set of 420 norms. After classifying them as permissions, obligations and prohibitions, there were 97 permissions, 164 obligations and 33 prohibitions. The remaining 126 are what has been called “environment”, which include properties such as the speed limit of a road, i.e., “The speed limit is 90 kilometers per hour on roads outside built-up areas”.

Each sentence was also mapped with the catalog of vocabularies provided by LOV. Sixty percent was considered the minimum percentage of mapped words that were necessary so that a norm could be understandable. Considering Table 3, it can be concluded that 328 sentences were valid for the research, which was approximately 78% of the possible norms. A sentence is considered fully mapped when all the terms at the time of obtaining the mappings have a correspondence with a term in LOV. It should be noted that stop words were removed previously.

Table 3. Percentage of mapped sentences.

Percentage of mapped words	Number of norms
100%	5
90-99%	18
80-89%	97
70-79%	127
60-69%	81
<60%	92

Finally, a use case of how a norm is converted into an ontology is shown so that a computer can interpret it. First, the text is preprocessed, starting with the extraction of a paragraph from the XML document and splitting it into sentences, corresponding each to a norm. Once the norm is isolated, for example: “The driver must not exceed the

speed limit specified on the maximum speed sign", it needs to be tokenized into isolated words. In the next step, this set of words is mapped with LOV's vocabularies. The results show that in this case, 80% of the sentence could be mapped. In Table 4, it can be seen which words have a mapping, if they have been made at a morphologic or semantic level and with which vocabulary of LOV it was done. It should be noted that a word can be mapped to several vocabularies, so it was decided to include only the useful mapping, which is the more accurate, based on the authors' opinion.

Table 4. Examples of mapped sentences.

Word	Mapped term	Mapping level	Vocabulary
driver	driver	Morphologic	Uco
speed	speed	Morphologic	Datex
limit	boundary	Semantic	Place
specified	condition	Semantic	Dqm
sign	sign	Morphologic	Semio

Apart from these mappings, there is a need to make a PoS analysis that is mapped with OLiA. The result can be seen in the following piece of code. In this case, each tag has its mapping with the PoS tags from Penn Treebank.

```
(ROOT
    (S
      (NP (DT The) (NN driver))
        (VP
          (MD must)
          (RB not)
          (VP
            (VB exceed)
            (NP
            (NP (DT the) (NN speed) (NN limit))
              (VP
              (VBN specified)
                (PP
                (IN on)
                (NP (DT the) (JJ maximum) (NN speed) (NN
                sign)))))))
                  (..)))
```

At this point, the norm can be annotated using the LOV vocabulary and the PoS tag mappings. Additionally, we know the structure of the norm because it has been categorized with the deep learning classifier. This norm was considered a prohibition, so it could be manually transformed in a SWRL rule, as can be seen in the following piece of code.

```
Declaration(Class(:NN))

Declaration(Class(:driver))

Declaration(Class(:NP))

Declaration(Class(:speed_limit))

SubClassOf(:NN:driver)

SubClassOf(:NP:speed_limit)

Declaration(Class(:VBN))

Declaration(ObjectProperty(:hasSpecified))

SubClassOf(:VBN:hasSpecified)

SubClassOf(:NN:sign)

sign(?x),     speed_limit(?y),     hasSpecified(?x,?y),
driver(?z) -> not_exceed(?z,?y)
```

5 Conclusions and future work

During this research, several issues were addressed. A tool able to extract traffic norms from an XML document was developed, mapping these norms to a catalog of vocabularies called LOV and to an ontology describing PoS tags. By obtaining these mappings, extra knowledge could be added to the terms by annotating them with LOV vocabularies. By obtaining this representation of the norms, they can be made understandable by machines. Additionally, PoS tags given by the OLiA ontology help to assign the function of each word in the sentence.

LOV mappings were built with two perspectives. The first, from a morphological point of view, shows that two terms are the same if they are written in the same way. The second mapping, from a semantic view, establishes a correspondence between two terms if they have the same meaning. For that purpose, the synonyms of the word are obtained to be mapped. Then, the synonyms are compared to all terms of the vocabularies, setting a mapping if they are the same word.

Additionally, a set of conditions was defined related to the structure of the sentence. Using these conditions, a classification of norms was developed, distinguishing between permissions, obligations and prohibitions. This classification was automatized with the help of a deep learning model.

Finally, a use case was developed by going through all the steps, from the raw text to the final ontology expressed with SWRL and the vocabularies given by the mappings.

In future works, some improvements can be made. N-grams can be used to make mappings not only with simple words but also with multiword expressions (words compounded of more than a word, i.e., “traffic light”). The process of choosing the most accurate vocabulary was performed manually. The process of generating SWRL rules can also be automatized as deep learning models, which can be used to find patterns. Once the process is as automatic as possible, created ontologies can be integrated with a multiagent system, thus allowing for experiments that could prove its application in the field. For that purpose, some real-use cases can be solved.

Acknowledgements

The work providing these results has received funding with Dora Plus Action scholarship from Tallinn University of Technology in Estonia.

References

1. Conceição, L. & Rossetti, R.J.: Multivariate modelling for autonomous vehicles: Research trends in perspective. Conference on 19th International Intelligent Transportation Systems (ITSC) 2016, IEEE, pp.83-87 (2016).
2. Du, J. & Barth, M.J.: Next-Generation Automated Vehicle Location Systems: Positioning at the Lane Level. IEEE Transactions on Intelligent Transportation System 9(1), 48-57 (2008).
3. Ahuja, A., Gautam, A. & Sharma.: Driverless Vehicles: Future Outlook on Intelligent Transportation. International Journal of Engineering Research and Application 5(2), 92-96 (2015).
4. Blyth, P., Mladenović, M., Nardi, B., Su, N., & Ekbia, H.: Driving the self-driving vehicle: Expanding the technological design horizon. In Proceedings of the International Symposium on Technology and Society Technical Expertise and Public Decisions, IEEE, pp.83-87 (2015).
5. Tsugawa, S., Yatabe, T., Hirose, T. & Matsumoto, S.: An Automobile with Artificial Intelligence. In: B. G. Buchanan (ed.), Proceedings of the 6th International Joint Conference on Artificial Intelligence, vol. 2, pp. 893-895 (1979).
6. Fisher, A.: Inside Google’s Quest to Popularize Self-Driving Cars. Popular Science, (2013).
7. Broggi, A., Cerri, P., Felisa, M., Laghi, M.C., Mazzei, L., & Porta, P.P.: The VisLab Intercontinental Autonomous Challenge: an extensive test for a platoon of intelligent vehicles. International Journal of Vehicle Autonomous Systems, 10(3), 147–164 (2012).
8. Clements, L. & Kockelman, K.: Economic Effects of Autonomous Vehicles. Transportation Research Record, (2606), 106-114 (2017).
9. Täks, E.: And Automated Legal Content Capture and Visualization Method. Thesis on Informatics and System Engineering. TUT Press, Tallinn (2003).
10. Wagner, A. & Cacciaguidi-Fahy, S.: Obscurity and clarity in the law: prospects and challenges. International Clarity Conference. Ashgate Publishing Ltd (2008).
11. Brighi, R., Lesmo, L., Mazzei, A., Palmirani, M., & Radicioni D. P.: Towards Semantic Interpretation of Legal Modifications through Deep Syntactic Analysis. In: Enrico Francesconi, Giovanni Sartor, and Daniela Tiscornia (eds.) Proceedings of the Conference on Legal Knowledge and Information Systems: JURIX 2008, IOS Press, Amsterdam, The Netherlands, 202-206 (2008).

12. Sergot, M. J. F. Sadri, F., Kowalski, R. A., Kriwaczek, F., Hammond, P. & Cory, H. T. 1986. The British Nationality Act as a logic program. Commun, ACM 29(5), 370-386 (1986).
13. Guda, V., Srujana, I. & Naik, M.V.: Reasoning in legal text documents with extracted event information. International Journal of Computer Applications, (28)7, 8-13 (2011).
14. R. Gruber, T.: A Translation Approach to Portable Ontologies. Knowledge Acquisition, 5(2), 199–220 (1993).
15. Lin, S. & Yen, E. Managed Grids and Cloud Systems in the Asia-Pacific Research Community. Springer Publishing Company, New York (2004).
16. Maedche, A. & Staab, S. Ontology Learning for the Semantic Web. IEEE Intelligent Systems, 16(2), 72-79 (2002)
17. Linked Open Vocabularies, http://lov.okfn.org/dataset/lov
18. Pavel, S. & Euzenat, J.: Ontology Matching: State of the Art and Future Challenges. IEEE Transactions on Knowledge and Data Engineering, 25(1), 158-176 (2013).
19. Bengio, Y., Courville, A.C., Goodfellow, I.J., & Hinton, G.E.: Deep Learning. Nature, 521(7553), 436-44 (2015).
20. Valente, A.: A Functional Ontology of Law. Artificial Intelligence and Law, (7), 341-361 (1994).
21. Soria, C., Bartolini, R., Lenci, A., Montemagni, S. & Pirrelli, V. (2007). Automatic Extraction of Semantics in Law Documents. In C. Biagioli, E. Francesconi, and G. Sartor (eds.), Proceedings of the V Legislative XML Workshop, pp. 253–266. European Press Academic Publishing (2007).
22. Fernandez S., Ito T. & Hadfi R.: Architecture for Intelligent Transportation System Based in a General Traffic Ontology. In: Lee R. (eds) Computer and Information Science 2015. Studies in Computational Intelligence, vol. 614, pp. 43-55. Springer, Cham. (2016).
23. Morignot, P., & Nashashibi, F.: An Ontology-Based Approach to Relax Traffic Regulations for Autonomous Vehicle Assistance. 12th IASTED International Conference on Artificial Intelligence and Applications, pp. 11-13 (2013).
24. Hülsen, M., Weiss, C., & Zöllner, J.M.: Traffic intersection situation description ontology for advanced driver assistance. IEEE Intelligent Vehicles Symposium (IV*)*, pp. 993-999 (2011)
25. Ou, W. & Elsayed. A.: Semantic processing for text mapping onto information space. In: J. M. Spector, D. G. Sampson, T. Okamoto, Kinshuk, S. A. Cerri, M. Ueno & A. Kashihara (eds.) 7th IEEE International Conference on Advanced Learning Technologies (ICALT 2007), pp. 931-932 (2007).
26. Ayush, S. & Jaideep, S.: Semantic Tagging for Documents Using 'Short Text' Information. International Journal of Computer Science & Information Technology, 6(4), p. 337 (2004).
27. Grobelnik, M., & Mladenic, D.: Mapping Documents onto Web Page Ontology. In: Berendt B., Hotho A., Mladenič D., van Someren M., Spiliopoulou M., Stumme G. (eds) Web Mining: From Web to Semantic Web, EWMF 2003, vol. 3209 (2016).
28. Kantarcioglu, M., Rachapalli, J., & Thuraisingham, B.M. (2012). Tag-based Information Flow Analysis for Document Classification in Provenance. In U. A. Acar & T. J. Green (eds.), 4th USENIX Workshop on the Theory and Practice of Provenance. (2012)
29. Gyawali, B., Shimorina, A., Gardent, C., Cruz-Lara, S. & Mahfoudh, M.: Mapping Natural Language to Description Logic. In: E. Blomqvist, D. Maynard, A. Gangemi, R. Hoekstra, P. Hitzler & O. Hartig (eds.), ESWC, vol. 1, pp. 273-288. (2017).
30. Zhang, X. & LeCun, Y. Text Understanding from Scratch. CoRR, abs/1502.01710. (2015).

31. Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., Gerber, M. S. & Barnes, L.E.: HDLTex: Hierarchical Deep Learning for Text Classification. CoRR, abs/1709.08267. (2017)
32. Liu, P., Qiu, X. & Huang, X.: Recurrent Neural Network for Text Classification with Multi-Task Learning. CoRR, abs/1605.05101. (2016)
33. Du, J., Gui, L., Xu, R. & He, Y.: A Convolutional Attention Model for Text Classification. In: X. Huang, J. Jiang, D. Zhao, Y. Feng & Y. Hong (eds.), 6th CCF International Conference, vol. 10619, pp. 183-195, (2018).
34. Parundekar, R.: Classification of Things in DBpedia using Deep Neural Networks. CoRR, abs/1802.02528. (2018)
35. Lenci, A., Montemagni, S., Pirrelli, V. & Venturi, G.: Ontology learning from Italian legal texts. In: Joost Breuker, Pompeu Casanovas, Michel C. A. Klein, and Enrico Francesconi (eds.), Proceedings of the 2009 conference on Law, Ontologies and the Semantic Web: Channelling the Legal Information Flood, pp. 75-94. IOS Press, Amsterdam, The Netherlands. (2009).
36. Tang, J., fung Leung, H., Luo, Q., Chen, D. & Gong, J.: Towards Ontology Learning from Folksonomies. In: C. Boutilier (ed.), Proceedings of the 21st International Joint Conference on Artificial Intelligence, pp. 2089-2094. (2009).
37. Jiang, X. & Tan, A.H.: CRCTOL: A semantic-based domain ontology learning system. Journal of the American Society for Information Science and Technology, 61(1), 150-168 (2009).
38. Langa, S.F., Sure-Vetter, Y., & Völker, J.: Supporting the Construction of Spanish Legal Ontologies with Text2Onto. In: P. Casanovas, G. Sartor, N. Casellas & R. Rubino (eds.), Computable Models of the Law, Languages, Dialogues, Games, Ontologies, vol. 4884, pp. 105-112. Springer (2008).
39. Casteleiro, M.A., Demetriou, G., Diz, J.J., Keane, J.A., Maroto, N., Maseda-Fernandez, D., Nenadic, G., Prieto, M.J., Read, W.J., & Stevens, R.: Ontology Learning with Deep Learning: A Case Study on Patient Safety Using PubMed. In: A. Paschke, A. Burger, A. Splendiani, M. S. Marshall & P. Romano (eds.), Proceedings of the 9th International Conference Semantic Web Applications and Tools for Life Sciences. (2016).
40. Buitelaar, P., Cimiano, P., Magnini, B.: Ontology learning from text: Methods, evaluation and applications. Frontiers in Artificial Intelligence and Applications, series 123. (2005).
41. Chiarcos, C. & Sukhareva, M. OLiA - Ontologies of Linguistic Annotation. Semantic Web 6(4), 379-386. (2015).
42. Fung, S. Y. C. & Watson-Brown, A.: The Template: A Guide for The Analysis of Complex Legislation, Institute of Advanced Legal Studies, Research Working Papers. (1994).
43. Klein, D. & Manning, C.D.: Accurate Unlexicalized Parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics, vol. 1, pp. 423-430. (2003).
44. Santorini, B.: Part-of-speech tagging guidelines for the Penn Treebank Project. Department of Computer and Information Science, University of Pennsylvania. (1990).
45. Fellbaum, C.: WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press. (1998)
46. Baumfalk, J., Poot, B., Testerink, B., & Dastani, M.: A SUMO Extension for Norm based Traffic Control Systems. In: Proceedings of the SUMO2015 Intermodal Simulation for Intermodal Transport, pp. 63-83. (2015).
47. Kim, Y.: Convolutional neural networks for sentence classification. CoRR, abs/1408.5882. (2014).

48. Lister, K.; Sterling, L.; Taveter, K. (2006). Reconciling Ontological Differences by Assistant Agents. Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-06): Future University, Hakodate, Japan, May 8-12, 2006. Ed. P. Stone; G. Weiss. ACM, 943-945. 10.1145/1160633.1160800/