Expert Systems With Applications 211 (2023) 118667

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Extracting Decision Model and Notation models from text using deep learning techniques☆

Alexandre Goossens *, Johannes De Smedt, Jan Vanthienen

KU Leuven, Leuven Institute for Research on Information Systems (LIRIS), Naamsestraat 69, 3000 Leuven, Belgium

ARTICLE INFO

Keywords:
Deep learning
Decision Model and Notation
DMN
Decision model extraction

ABSTRACT

Companies and organizations often use manuals and guidelines to communicate and execute operational decisions. Decision Model and Notation (DMN) models can be used to model and automate these decisions. Modeling a decision from a textual source, however, is a time intensive and complex activity hence a need for shorter modeling times. This paper studies how NLP deep learning techniques can extract decision models from text faster. In this paper, we study and evaluate an automatic sentence classifier and a decision dependency extractor using NLP deep learning models (BERT and Bi-LSTM-CRF). A large labeled and tagged dataset was collected from real use cases to train these models. We conclude that BERT can be used for the (semi)-automatic extraction of decision models from text.

1. Introduction

Certain types of decisions made by organizations are repetitive, e.g., calculating insurance rates. Individually these operational decisions have a small impact, but due to their high volume they have a significant impact on an organization or company (Vanthienen, 2021). To automate these operational decisions, companies have to formalize these using guidelines or manuals (Vanthienen, 2021). The Object Management Group (OMG) introduced the Decision modeling and Notation (DMN) that models both decision logic and decision dependencies (structure of a decision) (OMG, 2015) to better understand and automate these decisions. This modeling task however is time consuming and difficult due to the complex decision logic that needs to be understood (Vanthienen, 2021). Automated decision models extraction approaches from structured data have been proposed such as in Bazhenova and Weske (2016b) for process models and in Bazhenova, Bülow, and Weske (2016a), De Smedt, Hasić, vanden Broucke, and Vanthienen (2019) for logs. Unstructured textual sources however contain more decision logic information compared to logs or process models and have been studied to extract decision dependencies (Goossens, Claessens, Parthoens, & Vanthienen, 2021b) and decision logic (Arco et al., 2021) using natural language patterns. More sophisticated Natural Language Processing (NLP) deep learning techniques have been introduced such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin, Chang, Lee, & Toutanova, 2018) or Bidirectional LSTM with a Conditional Random Field (Bi-LSTM-CRF) (Huang, Xu, & Yu, 2015) showing promising results that NLP could help to automate decision model extraction from text (Li, Sun, Han, & Li, 2020; Lopez & Kalita, 2017; Otter, Medina, & Kalita, 2020; Torfi, Shirvani, Keneshloo, Tavaf, & Fox, 2020). Deep learning techniques do not need a predetermined pattern to understand a sentence for NLP tasks. Even though deep learning techniques exist for sentence classification or Named Entity Recognition (NER), to the best of our knowledge these have not yet been studied within the context of decision model extraction from text.

This paper investigates the use of two state-of-the-art deep learning techniques for decision model extraction from text for the first time, namely BERT and Bi-LSTM-CRF (Torfi et al., 2020) and is an extension of our very limited earlier research (Goossens, Claessens, Parthoens, & Vanthienen, 2021a; Goossens et al., 2021b). This paper extends the previous works by providing a complete implementation overview of the decision model extraction pipeline. Moreover, the dataset has been extended with more than 230 real-life sentences to evaluate the classification and decision dependency extraction steps of the pipeline. The deep learning models have also been hyper-optimized with a grid search which had not been done in the previous works. This work also performs a complete evaluation from text to DMN model on a few real-life example which has not been done in the previous works.

☆ This work was supported by the Fund for Scientific Research Flanders (project G079519N) and KU Leuven Internal Funds (project C14/19/082).
* Corresponding author.
E-mail addresses: alexandre.goossens@kuleuven.be (A. Goossens), johannes.desmedt@kuleuven.be (J. De Smedt), jan.vanthienen@kuleuven.be (J. Vanthienen).
[1] https://dmcommunity.org

https://doi.org/10.1016/j.eswa.2022.118667
Received 8 February 2022; Received in revised form 8 August 2022; Accepted 21 August 2022
Available online 26 August 2022

The main part of this research is focused on extracting the decision structure from text. For this, the performance of BERT and Bi-LSTM-CRF is compared. For the task of logic extraction, BERT is implemented. To perform these two tasks effectively, a first binary sentence classification task is performed on decision descriptions to identify which sentences are most relevant for decision modeling (Goossens et al., 2021a). A second binary classification task is performed on relevant sentences to classify sentences describing decision structure or logic. For the classification task, the performance of non-deep learning techniques is compared to the performance of BERT. A manually labeled dataset of 809 sentences from regulations, guidelines, decision modeling community challenges[1] and previous research (Etikala, Van Veldhoven, & Vanthienen, 2020) has been collected for this research resulting in a realistic corpus that organizations and companies are also confronted with. Lastly, a decision model extraction tool was built using deep learning techniques allowing for (semi)-automatic decision model extraction from text. As such the main contributions of this work are:

- The first complete pipeline to extract DMN models from text using deep learning is introduced.
- The very first labeled and tagged dataset to perform decision model extraction from text is made available to the research community.[2]
- The very first decision model extraction tool from text using deep learning is provided to the community. [2]

This paper is structured as follows: Section 2 introduces related work whilst Section 3 deals with the preliminaries and with the research questions. In Section 4, the methodology is laid out. Next, the experiments on individual sentences for decision dependency extraction are presented in Section 5. The experiments for the sentence classification task are presented in Section 6 whilst the experiments on full examples and the decision modeling tool are laid out in Section 7. Afterwards, the findings are discussed in Section 8 together with the limitations and future work. Finally, Section 9 concludes this paper.

2. Related work

OMG introduced DMN as a tool to model, communicate and execute operational decisions (OMG, 2015). Decision model extraction from various sources is a rather novel field. The authors of Bazhenova and Weske (2016b) studied extracting decision models from process models and from event logs (Bazhenova et al., 2016a; De Smedt et al., 2019). The aim of pattern-based approaches is to analyze possible formulations of sentences and categorize these formulations in so called patterns. The idea is that enough patterns are identified so that a wide array of formulations are covered by the approach. The next step is then to identify where the relevant elements for decision modeling can be found in each pattern. A pattern-based approach to extract decision tables from structured texts was studied in Kluza and Honkisz (2016) and from single sentences in Arco et al. (2021). Pattern-based approaches were also used to extract decision dependencies from unstructured texts (Etikala et al., 2020) and or the extraction of DMN models (Quishpi, Carmona, & Padró, 2021). Extracting Business Process Model and Notation (BPMN) models from text is a related field with textual process descriptions being sequential which decision descriptions are not. To extract BPMN models from text, pattern-based approaches were studied in de AR Goncalves, Santoro, and Baiao (2009), Friedrich, Mendling, and Puhlmann (2011), Sinha and Paradkar (2010) and one approach investigates the use of deep learning techniques (Han et al., 2020). The use of deep learning to extract decision dependencies or DMN tables from text has not been studied yet.

Rules extraction from text is a more mature and related field of decision model extraction from text dealing with finding individual rules in smaller texts. It is not concerned with finding general decision structures which are needed to understand the complex decision logic present in texts. One of the first approaches was to analyze a text on a syntactic level (Riloff, 1996; Soderland, 1999) which works if documents are following strict lay-out and sentence structures (Wyner & Peters, 2011). Yet, previous approaches are not that robust against sentence structure variations. This is why dependency trees were investigated (Lin & Pantel, 2001) and used to extract Semantics of Business Vocabulary and Business Rules (SBVR) vocabularies from Unified Modeling Language (UML) use cases (Danenas, Skersys, & Butleris, 2020). Extracting information from legal documents is investigated in Dragoni, Villata, Rizzi, and Governatori (2016) and the usage of deep learning has been studied within that context in Sansone and Sperlí (2021). To our knowledge, the use of deep learning to extract IF–THEN rules from texts has not been studied yet. Whilst artificial neural networks are researched in NLP (Alshemali & Kalita, 2020; Collobert & Weston, 2008; Yin, Kann, Yu, & Schütze, 2017), it is especially with the introduction of BERT that deep learning improved NLP tasks (Devlin et al., 2018). Since then BERT adaptations for other languages (Martin et al., 2019; Scheible, Thomczyk, Tippmann, Jaravine, & Boeker, 2020) or whole other transformer models such as GPT-2 (Radford et al., 2019) were developed. The field of deep learning for NLP is developing rapidly with models being fine-tuned for different NLP tasks regularly (Liang, Sun, Sun, & Gao, 2017; Young, Hazarika, Poria, & Cambria, 2018).

3. Preliminaries and research questions

In the following section, DMN is introduced in Section 3.1 then a motivation example in Section 3.2 followed by the definitions in Section 3.3. The research questions are elicited in Section 3.4.

3.1. DMN

The DMN standard is used to communicate and execute decisions and will be explained using a text based on the Belgian corona vaccination strategy also used in Goossens et al. (2021a).

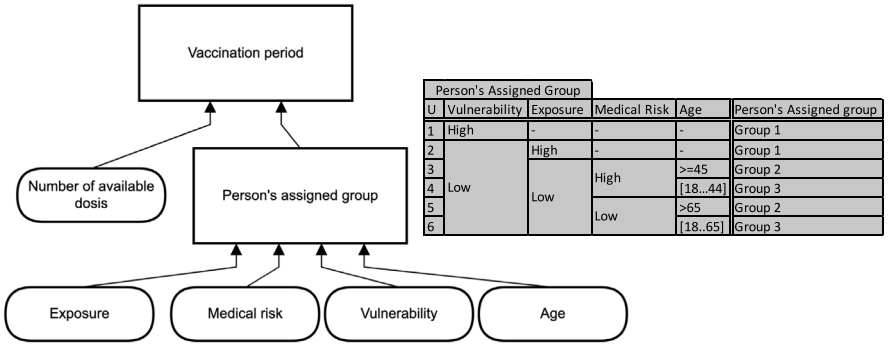
Corona Vaccination Example In order to expand the immunity of the population and protect the most vulnerable people against the Covid-19 virus, the vaccination must take place over different periods. The period in which a person will get vaccinated depends on the number of available doses and the person's assigned group. The person's assigned group is divided into three groups depending on their vulnerability, exposure, medical risks and age. If you are a resident or employee in a residential care center or if you work in a first line care occupation then you belong to the most vulnerable and exposed people and you will be vaccinated first. Moving on to the second group, if you are older than 65, or if you are between 45 and 65 with an increased medical risk due to healthcare issues, you will get vaccinated next. Lastly, the third group consists of the broader population of people above the age of 18. Invitations will be sent via text message, letter or email. After receiving a personal vaccine invitation, you need to register in order to confirm or move the appointment.

Fig. 1 shows the DMN model of the example.

A DMN model consists of two levels. The first level is called the decision requirements level and visualizes the structure of a decision in a Decision Requirements Diagram (DRD). A DRD is used to visualize the dependencies of decisions. In Fig. 1, rectangles represent decisions, e.g., "Vaccination period". Input data elements are represented with rounded rectangles (e.g., "Age") serving as decision inputs. Relations between input information items and decisions are represented with solid arrows or information requirements.

The second part of the DMN model is called the decision logic level. To communicate the decision logic, a decision table is often used. Fig. 1 shows the decision table of **Person's assigned group**. A decision table also has a hit policy which determines under which conditions decision rules are executed. A unique hit policy means for each case only 1 rule is fired.

[2] https://github.com/Goossens496/Extracting-DMN-models-from-text



Person's Assigned Group

U	Vulnerability	Exposure	Medical Risk	Age	Person's Assigned group
1	High	-	-	-	Group 1
2	Low	High	-	-	Group 1
3		Low	High	>=45	Group 2
4				[18...44]	Group 3
5			Low	>65	Group 2
6				[18..65]	Group 3

Fig. 1. DMN model.

3.2. Motivational example

In this section, we show how a DMN model is constructed from text using the **Corona Vaccination Example** from above.

In order to expand the immunity of the population and protect the most vulnerable people against the Covid-19 virus, the vaccination must take place over different periods. The first sentence is considered **irrelevant** since it does not describe decision logic or decision dependencies.

The period in which a person will get vaccinated depends on the number of available doses and the person's assigned group. This sentence does not contain any specific values or conditions but it does explain on which object attributes a decision depends on. We classify this sentence as **dependency**. The following dependencies are identified:

- Vaccination period **depends** on Number of available doses
- Vaccination period **depends** on Person's Assigned Group

The person's assigned group is divided into three groups depending on their vulnerability, exposure, medical risks and age. This sentence describes a **dependency** and the next dependencies are identified:

- Person's Assigned Group **depends** on exposure
- Person's Assigned Group **depends** on medical risk
- Person's Assigned Group **depends** on vulnerability
- Person's Assigned Group **depends** on age.

If you are a resident or employee in a residential care center or if you work in a first line care occupation then you belong to the most vulnerable and exposed people and you will be vaccinated first. This sentence contains specific values and explains an IF–THEN decision rule. It is classified as a **logic** sentence. In this paper, a decision tool automatically identifies the following IF–THEN parts:

- **IF-PART:** *If you are a resident or employee in a residential care center or if you work in a first line care occupation*
- **THEN-PART:** *then you belong to the most vulnerable and exposed people and you will be vaccinated first.*

It happens that a **logic** sentence contains ELSE or EXCEPT parts, in that case the tool also identifies these. This sentence explains when a person is assigned to group 1 to get vaccinated. The following decision dependencies are derived:

- Person's Assigned Group **depends** on exposure
- Person's Assigned Group **depends** on vulnerability

Moving on to the second group, if you are older than 65, or if you are between 45 and 65 with an increased medical risk due to healthcare issues, you will get vaccinated next. This sentence is classified as **logic** with the following parts:

- **IF-PART:** *if you are older than 65, or if you are between 45 and 65 with an increased medical risk due to healthcare issues*
- **THEN-PART:** *you will get vaccinated next.*

The following dependencies are identified:

- Person's Assigned Group **depends** on medical risk
- Person's Assigned Group **depends** on age.

Lastly, the third group consists of the broader population of people above the age of 18. This sentence is classified as **logic** with the following parts:

- **IF-PART:** *above the age of 18.*
- **THEN-PART:** *Lastly, the third group consists of the broader population of people*

The following dependency is found: Person's Assigned Group **depends** on age.

Invitations will be sent via text message, letter or email. After receiving a personal vaccine invitation, you need to register in order to confirm or move the appointment. The last two sentences are both considered **irrelevant** as no more decision information is given.

The decision dependencies give the DRD of Fig. 1. The top decision "Vaccination Period" **depends** on "Available doses" and "Person's assigned group". "Person's assigned group" **depends** on "exposure", " medical risk", "vulnerability" and "age". It is called an intermediary decision as it is both an input and output.

Next, the decision table of Fig. 1 is derived from the IF–THEN parts, i.e., the first rule of the decision table states that IF "vulnerability" is high THEN "Person's assigned group" is 1. This is found in the following parts:

- **IF-PART:** *If you are a resident or employee in a residential care center or if you work in a first line care occupation*
- **THEN-PART:** *then you belong to the most vulnerable and exposed people and you will be vaccinated first.*

3.3. Definitions

In this section, definitions are introduced. A decision determines an attribute of an object (e.g. person.age) based on input data (e.g. person.birthyear). We will refer to an object attribute as a concept.

Definition 1. A **base concept** is an input data element of a decision or an output attribute of another decision that is used as input for a different decision.

Definition 2. A **derived concept** is an output attribute of a decision.

Intermediary decisions are both base and derived concepts at the same time.

Definition 3. Each relation between a derived and a base concept can be represented by the following **decision dependency tuple** (a, b, d) with a from the set of action verbs $\mathcal{ACT}$, b from the set of base concepts $\mathcal{BAS}$, d from the set of derived concepts $\mathcal{DER}$, e.g., (divided, vulnerability, person's assigned group).

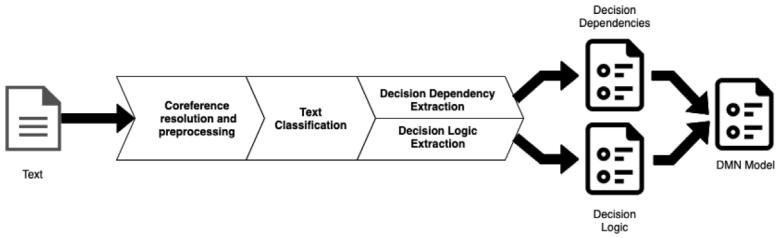


Fig. 2. Text to decisions.

Note that base concepts do not need to be mentioned before derived concepts in sentences, e.g., "Price is determined by discount".

Definition 4. Let $\mathcal{T}$ be a set containing t decision dependency tuples.

Definition 5. $\mathcal{T}$ is visualized in a DMN DRD ($\mathcal{I}$, $\mathcal{D}$, $\mathcal{R}$) with $\mathcal{I}$ a set of input information concepts, $\mathcal{D}$ a set of decisions and $\mathcal{R}$ a set of decision dependencies.

Definition 6. A **dependency** sentence is defined as a sentence describing the relation between concepts without the presence of values for these concepts.

Definition 7. A **logic** sentence is defined as a sentence that describes a decision rule consisting of an antecedent part (COND) and a consequence part (CONS).

Definition 8. Let $\mathcal{S}$ be a set of stop words containing 127 words provided by the NTLK package (Bird, Klein, & Loper, 2009). $\mathcal{S}$ contains words such as "you", "I", "if", "or".

The impact of removing stop words can either have a positive, negative or no impact on performances for NLP tasks (Ghag & Shah, 2015; Munková, Munk, & Vozár, 2013; Pradana & Hayaty, 2019) and needs to be investigated. Note that "if" and "or" describe the structure of a **logic** sentence.

3.4. Research questions

The motivational example gives rise to the following research questions:

1. Can (non-)deep learning techniques be used to build a classifier that classifies relevant sentences into **dependency** or **logic**?
2. Can deep learning techniques be used to identify set $\mathcal{T}$ from text?
3. For the sake of completeness, does removing stop words impact 2?
4. Is it possible to automatically create a DRD from an extracted set $\mathcal{T}$?
5. Can deep learning techniques identify different parts of a **logic** sentence?

4. Methodology

The general approach is introduced in Section 4.1. Next, dedicated sections will deal with coreference resolution Section 4.2, text classification in Section 4.3, decision dependency extraction in Section 4.4 and decision logic extraction in Section 4.5.

4.1. General approach

Fig. 2 represents the main steps to get a DMN model from a text. The following steps apply to both manual and automated approaches.

1. **Text Coreference Resolution & Preprocessing:** This step implements coreference resolution and preprocesses all sentences for further analysis. stop words are optionally removed for further research.
2. **Text Classification:** For this work we assume only relevant sentences are present as each sentence of the text is classified into **irrelevant for decision modeling** or **relevant** (Goossens et al., 2021a). A sentence is deemed **relevant** if it describes decision **logic** or **dependency**. For **relevant** sentences, some sentences describe the structure of a decision (**Dependency sentences**) and other sentences explain the exact decision logic (**Logic sentences**) (Quishpi et al., 2021). Hence, this paper will briefly investigate a classifier that classifies the relevant sentences into **Dependency** and **Logic**. As deep learning problems require a lot of data, transfer learning is implemented to gain extra knowledge from other pretrained models and not only count on a limited dataset (Devlin et al., 2018). In total 7 experiments were conducted for this problem.
3. **Decision Dependency and Logic Extraction on sentence level:** Decision dependency tuples are constructed for each sentence by identifying the relevant concepts and their relationships. Set $\mathcal{T}$ is created containing all the decision dependency tuples. The decision logic clauses are rendered in a table. In this paper, the decision dependency extraction is investigated, the decision logic extraction is shortly explored.
4. **Result of complete text:** From set $\mathcal{T}$ it is possible to derive a DRD. Texts can contain conflicts between concepts, e.g., "A depends of B. B depends of A". which is not allowed by the DMN standard. Hence, this tool can be used to deal with such conflicts and fix these manually. With the list of logic clauses, a decision table can be derived. With both a DRD and a decision table, a complete DMN model can be constructed.

4.2. Step 1: Coreference resolution and preprocessing

Coreference resolution is performed on the complete text and afterwards preprocessed. Coreference resolution is performed to identify the different words that refer to the same concept (e.g., period of vaccination = vaccination period) (Ng & Cardie, 2002). The NeuralCoref4 package[3] offers an automated approach for this. Next, all sentences are lowercased. In this paper the impact on the extraction of decision dependencies of removing stop words $\mathcal{S}$ is studied. Finally the sentences are tokenized meaning that a sentence is split into tokens with a token being a word, punctuation etc. This token is then put into a glossary of the corresponding pre-trained deep learning model.

[3] The package is introduced in the following blog: https://medium.com/huggingface/state-of-the-art-neural-coreference-resolution-for-chatbots-3302365dcf30 (accessed 2021-12-4).

4.3. Step 2: Text classification

In this step each relevant sentence of the text is classified as either **Dependency** or **Logic**. The dataset used for this problem has a unique label for each sentence. Note that certain sentences are not easy to unequivocally classify in a single label. The sentence *Inflation increases the price* describes logic and implies a dependency. It can be used to construct a decision table. We label it as **dependency**. Moreover, **Logic** sentences are more present in decision descriptions as decisions mainly describe decision logic. Only the classification of sentences into **Dependency** and **Logic** is evaluated in this paper. Common machine learning models will be compared to a deep learning model and the normal classification metrics are reported (precision, recall and F1-score). The models were trained and tested on a separate training set and test set.

4.3.1. Deep learning classifier

Currently, BERT (Devlin et al., 2018) is considered state-of-the-art for NLP tasks (Otter et al., 2020). Moreover, it allows for model-based transfer learning which means that a fine-tunable pre-trained model is provided as well as a dictionary containing features (the TokenID dictionary) and does not need to be trained from scratch. For this specific problem, BERT for Sequence Classification is used (Wolf et al., 2019) and a standard DistilBERT model (Sanh, Debut, Chaumond, & Wolf, 2019) consisting of 6 layers and 12 attention heads is fine-tuned. We chose DistilBERT because the authors of Sanh et al. (2019) conclude it is 40% smaller and 60% faster compared to BERT-base whilst losing only 3% in performance. DistilBERT is good enough to explore the use of deep learning for this classification task (Abadeer, 2020; Adoma, Henry, & Chen, 2020). This DistilBERT model will classify relevant sentences into **Dependency** or **Logic** and is provided with pretrained encoder weights which were trained on a big corpus before and are updated with backpropagation. A final single linear layer is added for the classification task.

4.3.2. Non-deep learning models:

Common machine learning models are also considered here with their higher understandability and lower computation times despite their lower performance. Two feature vector encoding methods were used:

1. **Bag of Words (BoW):** BoW counts the number of times a word is present in a document without looking at the context meaning that common words such as "the" or "you" are considered more important (higher weights) even though these do not provide meaningful information. BoW can also look at groups of n words also known as n-grams. For this problem, 2-grams was implemented.
2. **Term Frequence-Inverse Document Frequency (TF-IDF):** TF-IDF looks at the rarity of a word within a dataset on top of how often it occurs so that more weight will be given to more relevant words instead of common words.

In total, three common machine learning models are analyzed: logistic regression, naive bayes and support vector machines (SVM). In total 7 techniques have been analyzed i.e. each machine learning technique with both BoW and TF-IDF and BERT.

4.4. Step 3A: Decision dependency extraction

In this section, decision dependency extraction is discussed. To train a deep learning model for this task, a dataset of sentences is tagged manually meaning that each word is labeled. Next, classifiers are trained to predict these tags. Lastly, the tags are used to derive the decision dependency tuples and construct a DRD for visualization. Since decision dependencies can be extracted from both **Dependency** and **Logic** sentences, it might be that this task performs better on **Dependency** or **Logic** sentences. This is why it has been evaluated on three datasets: one dataset containing all sentences, one dataset containing only **Dependency** or **Logic** sentences respectively. For **Logic** sentences however, no action verb is present. The impact of stop words removal was also studied.

4.4.1. Decision dependency tagging

Decision dependency extraction can be seen as a Named Entity Recognition (NER) problem with base concepts, derived concepts and action verbs being the named entities. A common approach is to use the inside-outside-beginning (IOB) tagging format (Ramshaw & Marcus, 1999). For irrelevant chunks of text, an O(utside) tag is given. Derived concepts, base concepts and action verbs are given the DER, BAS and ACT tags respectively. When a concept consists of several words (e.g., credit score) the first word of a concept is tagged with a B(eginning)-XYZ tag (with XYZ = BAS or DER) and the other words of the same concept are tagged with an I(nside)-XYZ tag (e.g., credit (B-DER) score (I-DER)). The following set of tags is used {B-BAS,I-BAS,B-DER,I-DER,B-ACT,O}.

4.4.2. Decision dependency tag extraction

To extract these decision dependency tags two techniques were investigated:

1. **BERT:** NER tasks are not incorporated by default in BERT, but a pre-trained BERT model can be fine-tuned for NER (Kjeldgaard & Nielsen, 2021) . For this task, a standard BERT-base-uncased model was used consisting of 12 layers and 12 attention heads (Devlin et al., 2018). Moreover, a grid search was also performed meaning that for each dataset 180 BERT models were trained.
2. **Bi-directional LSTM-CRF models:** Sometimes Bi-LSTMs are performing better than BERT models on small datasets indicating that transfer learning might not be a requirement for this task (Ezen-Can, 2020). Moreover, it is possible that for decision dependency extraction transfer learning impacts performance negatively (Rosenstein, Marx, Kaelbling, & Dietterich, 2005; Wang, Dai, Póczos, & Carbonell, 2019). Bi-LSTM-CRF models are trained from scratch and use future input features on top of using past input features allowing them better insights in sentences (Huang et al., 2015). The final CRF-layer optimizes the current tag prediction based on previous tags. As such, Bi-LSTM-CRF models are an interesting starting point. Since Bi-LSTM-CRF models do not have their own word embeddings, the pre-trained word embedding GloVe (Pennington, Socher, & Manning, 2014) is used for vector representations having the advantage of capturing global statistics as well as local statistics of a corpus.

4.4.3. Deriving decision dependencies from the extracted tags

Concatenation of Tags: Once all the tags have been predicted, a list of base concepts and derived concepts is obtained for each sentence by concatenating all tags referring to the same concept (all the I-XYZ tags which are linked to the same B-XYZ tag are merged).

Construction of Dependency Tuples: Finally, decision dependency tuples (a, b, d) are constructed for each sentence. For a whole text the following sets are extracted:

- set $\mathcal{ACT}$ with elements a labeled as ACT
- set $\mathcal{BAS}$ with concepts b labeled as base concepts
- set $\mathcal{DER}$ containing concepts d with as derived concepts.
- set $\mathcal{T}$ containing t dependency tuples.

The sets $\mathcal{BAS}$ and $\mathcal{DER}$ can overlap since a concept can be a base and a derived concept at the same time.

4.4.4. DRD construction

A DRD tuple $(\mathcal{I}, \mathcal{D}, \mathcal{R})$, with $\mathcal{I}$ a set of input information concepts, $\mathcal{D}$ a set of (intermediate) decisions and $\mathcal{R}$ a set of decision dependencies, is derived and visualized from set $\mathcal{T}$ extracted previously. The DRD is derived as follows:

- $\mathcal{I} = \{\forall b \in \mathcal{BAS} | b \notin (\mathcal{BAS} \cap \mathcal{DER})\}$
- $\mathcal{D} = \{\forall d \in \mathcal{DER}\}$
- $\mathcal{R} = \{\forall t \in \mathcal{T}\}$

4.5. Step 3B: Decision logic extraction

Decision logic extraction clauses are extracted from **Logic** sentences. When building a DMN model, we find it useful that the DRD is modeled first and that afterwards the decision logic is elicited. This is why this work introduces decision logic extraction with deep learning but does not focus on it.

The steps for decision logic extraction are similar to the steps of Section 4.4 where the same IOB-tag methodology is used. The following tags are used: {B-COND,I-COND,B-EXCE,I-EXCE,B-CONS,I-CONS,B-ELSE,I-ELSE,O}.

4.5.1. Decision logic tag extraction

A fine-tuned BERT model for NER tasks was used since it is considered one of the top deep learning techniques for NLP tasks which for the purposes of this study is enough (Abadeer, 2020; Adoma et al., 2020). BERT was given a problem specific dataset containing **Logic** sentences that have been tagged manually according to the tagging scheme described above. Next, the BERT model is trained to extract the conditional, exception and (else) consequence parts.

4.5.2. Tabular representation of conditional sentence structure

Once the IOB-tags have been extracted, the final steps can be executed:

Concatenation of Tags: The tokens are concatenated in a similar fashion as in Section 4.4 resulting in a list for each type of clause for each sentence.

Tabular Representation of Conditional Sentence Structure: Lastly, each sentence is represented in a row of a table and each column represents the subclause type. With this table it is possible to further derive a DMN decision table.

5. Experiments for decision dependency extraction

In the following section, the experiments for decision dependency extraction are presented. The collected dataset and its metadata are presented in Section 5.1. The results are reported in Section 5.2 for the experiments conducted on **all** sentences, **dependency** sentences and **logic** sentences.

5.1. Collected data

A representative dataset is build from these sources:

- sentences extracted from academic papers (Batoulis & Weske, 2018; Valencia-Parra, Parody, Varela-Vaca, Caballero, & Gómez-López, 2021)
- from laws (Australian building regulations,[4] Compilation of patient protection and affordable care act[5])
- from KU Leuven regulations[6]
- from DM community challenges[7]
- internet search using keywords based on DMN application domains described in Section 5.2 of the OMG specification (OMG, 2015): "student admission requirements", "house loan eligibility", "work visa eligibility", "travel restrictions UK", "corona vaccination conditions"
- dataset published by Etikala et al. (2020)

Lastly, sentence variations were constructed using synonyms and/or other patterns. If a sentence contained bullet points, these were removed and replaced by "and" or "or" depending of the context. Let Y and W be a derived concept, Z an intermediate concept, X and V a base concept, e.g., (Loan (Y) depends on salary (Z) which depends on job (X).) The dataset contains the following patterns: X=>Y; Y<=X; Y<=Z<=X; X=>Z=>Y; Y<=X,W<=V; X=>Y,V=>W. To avoid data leakage, sentences do not occur in both training and test set. This has been manually checked for all the sentences by the authors. As such the training and test set do not overlap and 20% of the training set was separated as a validation set.

Once the dataset was collected, each sentence was manually classified into **Dependency** or **Logic** (see Tables 1 and 2). In reality **Logic** sentences are used more often to explain a decision hence their higher proportion. Finally, all sentences were manually tagged as follows: DER for derived concepts, BAS for base concepts and ACT for action verbs. A more elaborate description of the tagging scheme is provided in Section 4.4. An overview of the number of tags for each dataset is provided in Tables 1 and 2 . Identifying a derived concept correctly will prove crucial for the elicitation of the decision structure since a derived concept often depends on several base concepts and thus occurs less often.

5.2. Results

In this section, the results for **all** sentences, **dependency** and **logic** sentences are reported. The results for both BERT and Bi-LSTM-CRF models are reported with and without stop words. To reduce the impact of non-deterministic factors, the average results and their standard deviations based on 5 runs are presented. It was decided not to perform cross validation as it was deemed more relevant to only test on sentences that are not variants from the training sentences. As such, the test set only contains unique sentences for which no variations have been made and that can be linked to a source. As such, the provided results show the results in case the deep learning model is confronted with totally new sentences. The precision, recall and F1-score are reported. A grid search was performed with the hyperparameter ranges given in Table 3. For each dataset and each model (with and without stop words), the model was trained 90 times with each time different hyperparameters and validated on the validation set. The best hyperparameters were selected to test the model on. The experiments were conducted within the Google colab pro environment. To optimize the BERT models, it takes approximately 2.5 h and to optimize the Bi-LSTM-CRF models, it takes approximately 5 h.

5.2.1. All sentences

An overview of the results is provided in Tables 4–6. When looking at the results in Tables 4 and 5, it is clear that the impact of removing stop words is negative as no metric is the best for BERT without stop words. A similar conclusion can be drawn when looking at the results for Bi-LSTM-CRF. When comparing Bi-LSTM-CRF with stop words and BERT with stop words, BERT with stop words is the best performing model for the extraction of decision dependency tags for all sentences. In short, BERT models are clearly better than Bi-LSTM-CRF models and stop words removal has a negative impact. The best hyperparameters are reported in Table 7.

[4] https://content.legislation.vic.gov.au/sites/default/files/2020-10/18-38sra013%20authorised.pdf

[5] http://housedocs.house.gov/energycommerce/ppacacon.pdf

[6] https://www.kuleuven.be/education/regulations/2021/?faculteit=50000102

[7] https://dmcommunity.org/challenge/

Table 1
Training set information.

Dataset	Training size	Mean # characters	B-DER	I-DER	B-BAS	I-BAS	B-ACT	All tags
All sentences								
With stopwords	577	114	648	1468	1050	1737	/	4903
Without stopwords	577	70	632	949	1032	1186	/	3799
Dependency sentences								
With stopwords	249	116	265	772	535	1154	253	2979
Without stopwords	249	76	252	498	520	777	253	2300
Logic sentences								
With stopwords	328	112	383	696	515	583	/	2177
Without stopwords	328	65	380	451	512	409	/	1752

Table 2
Test set information.

Dataset	Test size	Mean # of characters	B-DER	I-DER	B-BAS	I-BAS	B-ACT	All tags
All sentences								
With stopwords	232	114	266	435	409	899	/	2009
Without stopwords	232	69	247	316	380	614	/	1557
Dependency sentences								
With stopwords	55	113	62	101	91	142	49	445
Without stopwords	55	69	60	80	83	102	46	371
Logic sentences								
With stopwords	177	111	179	297	305	741	/	1522
Without stopwords	177	69	165	216	288	503	/	1172

Table 3
Parameters optimized through grid search.

Parameters	Values
Dropout rate	[0, 0.25, 0.5]
Epochs	[5,6,7,8,9]
Training batch size	[8,16,32]
Learning rate	[0.0001,0.00001]

It was decided to also investigate the impact of only removing the words "a", "an", "the" (see Table 5). Notice that the results for BERT with stop words and BERT without "a", "an", "the" are rather similar yet different in interesting ways. First, a small reminder that DER stands for derived concepts, BAS for base concepts and ACT for action verbs. The first word of a base concept is tagged with B-BAS and the other words of that same base concepts with I-BAS. BERT with stop words has a higher precision for B-DER and B-BAS, meaning it can identify the first word of a concept better than its counter part. Moreover, recall is also higher for I-DER and I-BAS for BERT with stop words. For BERT without "a", "an", "the" precision is higher for I-DER and I-BAS and recall is higher for B-DER and B-BAS. It has been decided to not experiment further without "a", "an", "the" because the authors deem that identifying the first words of a concepts is crucial for a good DMN model.

5.2.2. Dependency sentences

The results are shown in Tables 8 and 9. Looking at the results from BERT, removing stop words has a negative impact as no metric is the best for BERT without stop words. In fact, BERT without stop words predicts everything wrongly (see Table 8). This can be attributed to the fact that BERT labels every word with the O(utside) label. With the variability of the BERT models being reduced as much as possible within the test environment, all five BERT models reported the same results in this case. This does highlight the importance of stopwords for BERT to understand dependency sentences. Looking at the Bi-LSTM-CRF results, the negative impact of stop words removal is less obvious. Yet, both Bi-LSTM-CRF results are well below the results of BERT with stop words. The best hyperparameters are reported in Table 7.

5.2.3. Logic sentences

The results are provided in Tables 10 and 11. Once again, removing stop words has a negative impact as no metric is the best for BERT without stop words. As can be seen, the standard deviations for BERT without stop words equals 0, this can be attributed to the fact the variation of the BERT models has been reduced as much as possible. As such by removing the stopwords, the resulting BERT models across five runs learned the same and gave the same results hence reducing the standard deviations to 0. When looking at the Bi-LSTM-CRF results, a similar conclusion is drawn. Yet, both Bi-LSTM-CRF results are well below the results of BERT with stop words. The best hyperparameters are reported in Table 7.

6. Experiments for sentence classification

The following section reports the sentence classification results. The same labeled dataset as presented in Tables 1 and 2 for all sentences was used.

6.1. Results

The sentence classification results are provided in Table 12 for BoW, Table 13 for TF-IDF and Table 14 for BERT. Since logistic regression, naive bayes and support vector machines consistently reproduce the same results when given the same data, the standard deviations are not provided for these models. Furthermore, the implementation of DistilBERT allows for having consistent results for every run on the same data, as such the standard deviations are also not provided for BERT for text classification. Looking at Table 12, SVM is the best performing model even though logistic regression is not that far off. For Table 13, SVM is again the best performing model, logistic regression and naive bayes are both performing similarly. Comparing BoW with TF-IDF, the results are quite similar for both. Finally, when looking at Table 14, BERT is the best performing model at classifying sentences into **Logic** and **Dependency** compared to SVM which was the best technique for both BoW and TF-IDF.

Table 4
Results for all sentences: BERT.

Level	BERT without stop words			BERT with stop words		
	Precision	Recall	F1-score	Precision	Recall	F1-score
B-DER	0.5952 ± 0.0045	**0.5232** ± 0.0161	**0.5567 ± 0.0069**	**0.637 ± 0.0011**	0.487 ± 0.0349	0.5076 ± 0.0178
I-DER	0.6167 ± 0.0151	0.6286 ± 0.0028	0.6225 ± 0.0065	**0.7055 ± 0.0124**	**0.7595 ± 0.014**	**0.7313 ± 0.0001**
B-BAS	0.4937 ± 0.0049	0.5216 ± 0.0023	0.5073 ± 0.0037	**0.5931 ± 0.0074**	**0.5731 ± 0.0097**	**0.5829 ± 0.0086**
I-BAS	0.62 ± 0.0128	0.3614 ± 0.0118	0.4564± 0.0056	**0.7022 ± 0.012**	**0.4933 ± 0.0101**	**0.5793 ± 0.0031**
AVG-MICRO	0.5761 ± 0.0056	0.4791 ± 0.0084	0.5231 ± 0.0026	**0.6694 ± 0.0079**	**0.5793 ± 0.0011**	**0.6211 ± 0.0028**
AVG-MACRO	0.5796 ± 0.0109	0.5087 ± 0.0083	0.5357 ± 0.0024	**0.6594 ± 0.0077**	**0.6035 ± 0.0051**	**0.6262 ± 0.0058**

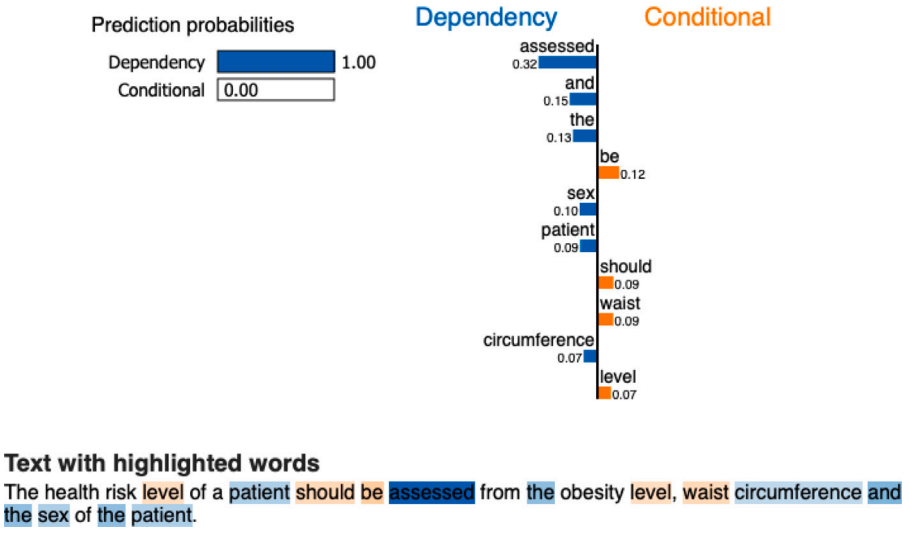


Fig. 3. Lime for dependency sentence.

Table 5
Results for all sentences BERT without A, An , The.

Level	BERT without A, An, The		
	Precision	Recall	F1-score
B-DER	0.5989 ± 0.0089	0.6045 ± 0.0083	0.6016 ± 0.0002
I-DER	0.7427 ± 0.0077	0.7085 ± 0.0032	0.7252 ± 0.0054
B-BAS	0.5547 ± 0.0041	0.6136 ± 0.0207	0.5826 ± 0.0117
I-BAS	0.739 ± 0.0037	0.4868 ± 0.0244	0.5866 ± 0.0172
AVG-MICRO	0.6682 ± 0.0031	0.5757 ± 0.0156	0.6184 ± 0.0079
AVG-MACRO	0.6577 ± 0.0016	0.6033 ± 0.0125	0.624 ± 0.0058

6.2. Understanding how BERT classifies text

With BERT being the best performing model, it is important for users to understand its reasoning and whether this matches their intuition. In Fig. 3, Local Interpretable Model Agnostic Explanations (LIME) values (Ribeiro, Singh, & Guestrin, 2016) are shown of how BERT classifies a sentence as **Dependency**. These LIME values indicate how much each word contributes to the classification of a sentence. In this case "assessed" contributes to the **Dependency** label as well as the substantives such as "sex" or "patient". If only concepts and no values are present a sentence can be labeled as **dependency**. Other words that make BERT classify a sentence as **dependency** are the presence of verbs that indicate a relationship between two words, e.g., "depends". When looking at how BERT labels a sentence as **logic** an interesting distinction is made. Firstly, words such as "if" or "then" contribute to the **logic** label. Secondly, whilst the presence of concepts increases the probability of being labeled as **dependency**, the presence of values for these concepts such as "high", "lower" or "bad" contributes to the **logic** label.

7. Experiments on full examples

The previous sections dealt with classification and decision dependency extraction on sentence level, this section will test DMN model extraction on 6 full real-life examples. For this, a DMN model extraction tool was constructed and is presented in Section 7.1. Next, the results of the extracted DRDs of real internet examples are reported in Section 7.2.

7.1. The decision model extraction tool

The DMN model extraction tool takes as input a decision description and automatically classifies sentences into **irrelevant** (for the decision model), **dependency** or **logic** (see Fig. 4). Next, it is possible to extract decision tuples and get a DRD graph using two BERT with stop words models for **dependency** or **logic** sentences respectively. The authors believe this solution might be more fine-tuned for the specificities of each type of sentence.

7.1.1. Step 1: Text analysis and decision dependency tuple construction

In the top left of Fig. 4, the original text must be provided directly in the text box or with a .txt-file. Next, the text can be classified by clicking on the "Classify text" button. This will make the original text appear again with a color scheme as well as the predicted label. Blue stands for **dependency** sentences and green stands for **logic** sentences. When clicking on the "Build DRD tuple" button, the decision tool extracts a list of concepts from both **dependency** and **logic** sentences and also extracts DRD tuples. The resulting lists are returned on the top right corner of the decision tool. By clicking on the button "Get DRD graph", a DRD is build. Note that decision dependencies and concepts appearing multiple times are identified automatically and only appear once in the final dependencies list. Finally, the prototype is also able to extract the

Table 6
Results for all sentences: Bi-LSTM-CRF.

Level	Bi-LSTM-CRF without stop words			Bi-LSTM-CRF with stop words		
	Precision	Recall	F1-score	Precision	Recall	F1-score
B-DER	0.3627 ± 0.0256	0.3117 ± 0.0497	0.3339 ± 0.037	**0.4692 ± 0.1064**	**0.4331 ± 0.0305**	**0.443 ± 0.0315**
I-DER	0.4344 ± 0.0149	0.4025 ± 0.0325	0.4176 ± 0.0232	**0.53 ± 0.0518**	**0.5384 ± 0.0308**	**0.5323 ± 0.0237**
B-BAS	0.2987 ± 0.018	0.4121 ± 0.027	0.3438 ± 0.0183	**0.3475 ± 0.0181**	**0.443 ± 0.0523**	**0.3876 ± 0.0163**
I-BAS	0.4081 ± 0.0286	0.172 ± 0.0274	0.2405 ± 0.0269	**0.5097 ± 0.029**	**0.279 ± 0.0568**	**0.3563 ± 0.0461**
AVG-MICRO	0.3615 ± 0.0124	0.2996 ± 0.0093	0.3275 ± 0.0089	**0.4538 ± 0.0273**	**0.3889 ± 0.0278**	**0.4183 ± 0.0217**
AVG-MACRO	0.376 ± 0.0148	0.3246 ± 0.0157	0.3346 ± 0.012	**0.4641 ± 0.0375**	**0.4187 ± 0.0116**	**0.4298 ± 0.0252**

Table 7
Best hyperparameters.

Hyperparameters for all sentences					
Model	Learning rate	Dropout rate	Epoch	Training batch size	Warm-up steps
BERT without stop words	0.0001	0.5	5	8	0
BERT with stop words	0.0001	0	7	8	0
BERT without a, an, the	0.0001	0.5	6	8	0
Bi-LSTM-CRF without stop words	0.0001	0.25	9	8	0
Bi-LSTM-CRF with stop words	0.0001	0	9	8	0
Hyperparameters for dependency sentences					
Model	Learning rate	Dropout rate	Epoch	Training batch size	Warm-up steps
BERT without stop words	0.0001	0.5	6	8	0
BERT with stop words	0,0001	0	6	8	0
Bi-LSTM-CRF without stop words	0.0001	0.25	9	8	0
Bi-LSTM-CRF with stop words	0.0001	0	8	8	0
Hyperparameters for logic sentences					
Model	Learning rate	Dropout rate	Epoch	Training batch size	Warm-up steps
BERT without stop words	0.0001	0.25	6	16	0
BERT with stop words	0.0001	0	9	8	0
Bi-LSTM-CRF without stop words	0.0001	0.25	8	8	0
Bi-LSTM-CRF with stop words	0.0001	0.25	8	8	0

Table 8
Results for dependency sentences: BERT.

Level	BERT without stop words			BERT with stop words		
	Precision	Recall	F1-score	Precision	Recall	F1-score
B-DER	0 ± 0	0 ± 0	0 ± 0	**0.6176 ± 0**	**0.6774 ± 0**	**0.6482 ± 0.0045**
I-DER	0 ± 0	0 ± 0	0 ± 0	**0.6308 ± 0.0172**	**0.8297 ± 0.0177**	**0.7164 ± 0.0051**
B-BAS	0 ± 0	0 ± 0	0 ± 0	**0.6207 ± 0.0056**	**0.7516 ± 0.0098**	**0.6799 ± 0.0074**
I-BAS	0 ± 0	0 ± 0	0 ± 0	**0.6756 ± 0.0061**	**0.8099 ± 0.0157**	**0.7367 ± 0.0102**
AVG-MICRO	0 ± 0	0 ± 0	0 ± 0	**0.6661 ± 0.0156**	**0.6939 ± 0.0456**	**0.6795 ± 0.0304**
AVG-MACRO	0 ± 0	0 ± 0	0 ± 0	**0.6447 ± 0.0071**	**0.7712 ± 0.004**	**0.7023 ± 0.0059**

Table 9
Results for dependency sentences: Bi-LSTM-CRF.

Level	Bi-LSTM-CRF without stop words			Bi-LSTM-CRF with stop words		
	Precision	Recall	F1-score	Precision	Recall	F1-score
B-DER	**0.4198 ± 0.0307**	0.2867 ± 0.0361	0.3399 ± 0.0313	0.3993 ± 0.0348	**0.4355 ± 0.0582**	**0.4129 ± 0.0175**
I-DER	0.4193 ± 0.0278	**0.585 ± 0.1044**	0.4712 ± 0.0189	**0.456 ± 0.069**	0.5703 ± 0.089	**0.4999 ± 0.0499**
B-BAS	**0.3732 ± 0.0292**	0.7084 ± 0.0101	**0.4883 ± 0.0261**	0.3411 ± 0.0273	**0.778 ± 0.0428**	0.4729 ± 0.0195
I-BAS	0.3296 ± 0.0121	**0.6569 ± 0.0601**	0.4379 ± 0.0121	**0.502 ± 0.039**	0.462 ± 0.0799	**0.4773 ± 0.0453**
AVG-MICRO	0.5282 ± 0.0288	0.3 ± 0.0238	0.3819 ± 0.0213	**0.5542 ± 0.0286**	**0.3592 ± 0.0233**	**0.4355 ± 0.0212**
AVG-MACRO	**0.4975 ± 0.01**	0.3502 ± 0.0127	0.3951 ± 0.0111	0.418 ± 0.0141	**0.5362 ± 0.0229**	**0.4697 ± 0.017**

Table 10
Results for logic sentences: BERT.

Level	BERT without stop words			BERT with stop words		
	Precision	Recall	F1-score	Precision	Recall	F1-score
B-DER	0.5247 ± 0	0.509 ± 0	0.5167 ± 0	**0.6035 ± 0.0341**	**0.565 ± 0.0538**	**0.581 ± 0.0084**
I-DER	0.6528 ± 0	0.4234 ± 0	0.5137 ± 0	**0.8034 ± 0.0095**	**0.6146 ± 0.0142**	**0.6965 ± 0.0127**
B-BAS	0.3975 ± 0	0.4315 ± 0	0.4138 ± 0	**0.5163 ± 0.0293**	**0.5119 ± 0.0201**	**0.5141 ± 0.0247**
I-BAS	0.549 ± 0	0.2951 ± 0	0.3839 ± 0	**0.7381 ± 0.0128**	**0.3846 ± 0.0046**	**0.5057 ± 0.007**
AVG-MICRO	0.5083 ± 0	0.3809 ± 0	0.4354 ± 0	**0.6705 ± 0.0183**	**0.4758 ± 0.0028**	**0.5565 ± 0.0046**
AVG-MACRO	0.531 ± 0	0.4148 ± 0	0.457 ± 0	**0.6653 ± 0.0166**	**0.5191 ± 0.0108**	**0.5743 ± 0.0026**

Table 11
Results for logic sentences: Bi-LSTM-CRF.

Level	Bi-LSTM-CRF without stop words			Bi-LSTM-CRF with stop words		
	Precision	Recall	F1-score	Precision	Recall	F1-score
B-DER	0.3582 ± 0.0412	0.4073 ± 0.0379	0.3783 ± 0.0169	**0.5276 ± 0.0518**	**0.4447 ± 0.0419**	**0.4795 ± 0.0142**
I-DER	0.308 ± 0.0315	0.1176 ± 0.0145	0.1693 ± 0.0146	**0.5946 ± 0.0528**	**0.3859 ± 0.0352**	**0.4652 ± 0.0128**
B-BAS	0.2278 ± 0.0202	0.4542 ± 0.0497	0.3016 ± 0.0139	**0.2755 ± 0.0255**	**0.5043 ± 0.0342**	**0.3546 ± 0.0126**
I-BAS	0.3314 ± 0.016	0.1105 ± 0.0118	0.1654 ± 0.0131	**0.5636 ± 0.0415**	**0.2346 ± 0.0277**	**0.3294 ± 0.0221**
AVG-MICRO	0.273 ± 0.0111	0.2381 ± 0.0113	0.2541 ± 0.0074	**0.4271 ± 0.0193**	**0.3428 ± 0.0219**	**0.3799 ± 0.0151**
AVG-MACRO	0.3064 ± 0.0095	0.2724 ± 0.0103	0.2501 ± 0.014	**0.4903 ± 0.033**	**0.3923 ± 0.021**	**0.4015 ± 0.0197**

Table 12
Text classification results :BoW.

	Logistic regression			Naive bayes			Support vector machines		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Conditional	**0.93**	0.88	0.9	0.9	**0.92**	**0.91**	**0.93**	0.89	**0.91**
Dependency	0.67	**0.8**	0.73	**0.72**	0.65	0.69	0.69	**0.8**	**0.74**
accuracy	/	/	0.86	/	/	0.86	/	/	**0.87**
macro avg	0.8	**0.84**	**0.82**	**0.81**	0.79	0.8	**0.81**	**0.84**	**0.82**
weighted avg	0.87	0.86	0.86	0.85	0.86	0.86	**0.88**	**0.87**	**0.87**

Table 13
Text classification Results: TF-IDF.

	Logistic regression			Naive bayes			Support vector machines		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Conditional	0.85	0.93	0.89	0.83	**0.95**	0.88	**0.92**	0.92	**0.92**
Dependency	0.68	0.49	0.57	0.69	0.36	0.48	**0.73**	**0.75**	**0.74**
accuracy	/	/	0.82	/	/	0.81	/	/	**0.88**
macro avg	0.73	0.71	0.73	0.76	0.66	0.68	**0.83**	**0.83**	**0.83**
weighted avg	0.81	0.82	0.81	0.79	0.81	0.79	**0.88**	**0.88**	**0.88**

Table 14
Text classification BERT vs Support Vector Machines (SVM).

	BERT			SVM + BoW			SVM + TFIDF		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Conditional	**0.96**	0.87	0.83	0.93	0.89	0.91	0.92	**0.92**	**0.92**
Dependency	**0.79**	**0.87**	**0.83**	0.69	0.8	0.74	0.73	0.75	0.74
accuracy	/	/	**0.91**	/	/	0.87	/	/	0.88
macro avg	**0.87**	**0.9**	**0.89**	0.81	0.84	0.82	0.83	0.83	0.83
weighted avg	**0.92**	**0.91**	**0.92**	0.88	0.87	0.87	0.88	0.88	0.88

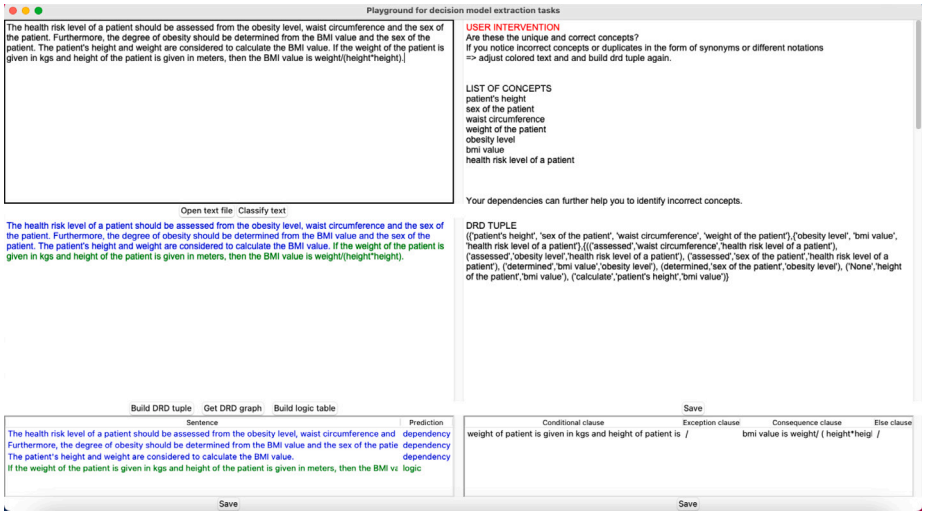


Fig. 4. BMI example with manual coreference resolution.

logical clauses of logic sentences. By clicking on the "Build logic table" button, the logic table is shown in the bottom right corner. Currently, coreference resolution cannot completely be solved automatically. The tool does not know that "degree of obesity" and "obesity level" are the same concept. This can be fixed manually by using the same words for the same concepts.

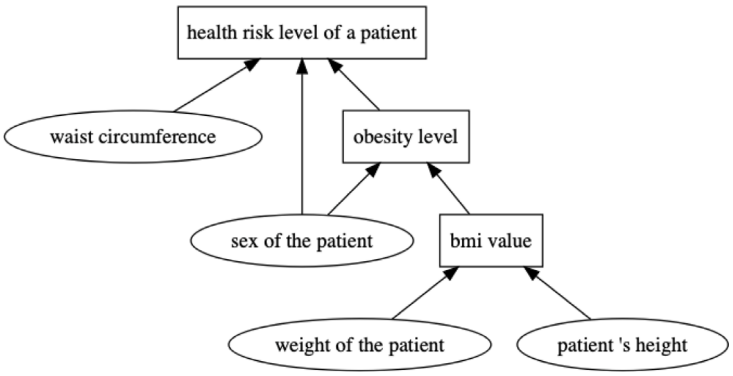


Fig. 5. Tool generated DRD after coreference resolution.

Table 15
Results for real examples.

Source		Decision and input data						Decision dependencies					
	#words	#gold	#pred	#ok	P	R	F1	#gold	#pred	#ok	P	R	F1
Fraud Rating Score	101	5	5	5	100%	100%	100%	4	4	4	100%	100%	100%
Housing Loan Eligibility	38	8	8	8	100%	100%	100%	7	7	7	100%	100%	100%
Obama Care	51	6	6	5	83%	83%	83%	5	8	4	50%	80%	62%
Personal Loan Eligibility	44	8	9	8	89%	100%	94%	7	8	7	88%	100%	93%
Student AID US	59	6	7	6	86%	100%	92%	5	6	5	83%	100%	91%
Student Support VUB	85	5	5	5	100%	100%	100%	4	4	4	100%	100%	100%
Total	378	38	40	37	93%	97%	95%	32	38	32	87%	97%	91%

7.1.2. Step 2: DRD construction

Based on the final decision dependency tuples, a DRD tuple (I, D, R) is generated by the tool (see Fig. 5). This can be done as the extracted decision dependency tuple are of the format (a, b, d). It is also possible to directly adapt the concepts and decision tuples in the DRD file to generate a correct DRD.

7.2. Experiments

In the following section, the experiments are presented. In Section 7.2.1, the collected examples are presented. In Section 7.2.2, the results on the real examples are reported with the runtimes reported in Section 7.2.3.

7.2.1. Collected examples

Full examples were collected from the DM community or using web searches using keywords in similar fashion as for the sentence collection. An example was included when a text was describing a decision structure or logic without using lay-out. When there were bullet points these were removed and connected by "and". One of the examples is the following: *To qualify for Obamacare subsidies you must meet the following criteria You are currently living in the United States and You are a US citizen or legal resident and You are not currently incarcerated and Your income is no more than 400% (or 500% in 2021 and 2022) of the FPL.* The collected examples deal with different eligibilities for loans or social support and one example deals with fraud rating score.

7.2.2. Results

For each example, we report the number of words (#words), number of decisions and input data items modeled by a human (#gold), number of decisions and input data items modeled by the tool (#pred) and number of correctly identified decisions and input data items (#ok) in Table 15. Precision (P= #ok/#pred), recall (R=#ok/#gold) and F1-score (F1= 2PR/(P+R)) are also reported. Similar metrics are reported for decision dependencies.

Table 16
Runtimes of real examples in milliseconds.

Source	#words	Text classification	DRD tuple construction
Fraud Rating Score	101	6067	11232
Housing Loan Eligibility	38	1371	3981
Obama Care	51	914	992
Personal Loan Eligibility	44	1094	2819
Student Aid US	59	2141	7163
Student Support VUB	85	1948	4347

7.2.3. Runtimes

For each example, the runtimes in ms for text classification and DRD tuple construction are reported (Table 16). Even on a slow machine with 4 GB RAM and 1600 MHz Intel double core i5, the tasks were performed reasonably fast.

8. Discussions

The resulting findings are discussed in Section 8.1. The limitations and future work are discussed in Section 8.2

8.1. Discussion of research findings from results

8.1.1. Text classification

Text classification Section 4.3 is beneficial for two reasons. First, the decision dependency extractors are trained on each type of sentence learning the specificities of those sentences even if not all sentences clearly belong in either **dependency** or **logic**. Second, it follows the human way of modeling which in the case of semi-automatic modeling is important to users as it allows for a timely intervention that matches their own modeling intuition. We conclude that BERT is the best performing model at classifying sentences into **dependency** or **logic**. Interestingly, BERT uses stop words, adjectives and substantives to classify a sentence just like a human modeler would. Currently, this is the only work to explicitly propose a text classification of sentences in the context of decision modeling. As such it is not possible to perform a comparison with state-of-the-art techniques.

8.1.2. Decision dependency extraction

For decision dependency extraction, the performance of BERT and Bi-LSTM-CRF on decision dependency tagging was studied. Our findings conclude that BERT is the best model for this task (see Tables 12–14). Once again, this confirms why BERT is considered state-of-the-art even though (Ezen-Can, 2020) suggest that Bi-LSTM-CRF could perform better than BERT on a small dataset. It might be that the structure of BERT is better equipped to "understand" a task such as decision dependency tagging which inherently is a complex modeling problem even for humans. Moreover, BERT has an in-built advantage of making use of transfer learning which Bi-LSTM-CRF models do not have.

The impact of stop words removal was also studied and the results show that removing stop words has a negative impact meaning that BERT uses stop words to understand sentence structures. Interestingly, extracting decision dependencies from **dependency** or **logic** sentences does not yield very different results. Whilst the extracting dependencies from **logic** sentences seems to perform slightly better than for **dependency** sentences, it important to note that **logic** sentences follow a more predictable pattern than **dependency** sentences.

Lastly, BERT seems to have issues with identifying the first words of a concept (B-tags) compared to the consecutive words of the same concept (I-tags) due to Precision being higher for the I-tags compared to the B-tags in Tables 4, 8 and 10. This matches with the challenge of identifying a concept in the first place not identifying the words that belong with a concept. Similar to the text classification task, this is the only work that explicitly extracts decision dependencies in a measurable way (e.g. with a tagging scheme). Therefore, it is not possible to perform a comparison with state-of-the-art techniques.

8.1.3. Full examples

First, the results of the Obama care example (see Table 15) are discussed. In this example six concepts should be identified: {Determine eligibility for Obama care subsidies, living place, US citizen status, legal resident status, incarceration status, income}. Only the first concept should be identified as a decision and the other 5 concepts should be considered as input data. The tool however predicts that income is a decision as well. Hence, why 5 out 6 concepts are predicted correctly. Regarding the decision dependencies, the 4 remaining concepts that are predicted as input data for *Determine eligibility for Obama care subsidies* are also predicted to be input data for *income*. Hence, there are 4 dependencies that are incorrect. This explains why the Precision and F1-score are lower for decision dependencies. However, the tool did find all the other dependencies between the input data and the decision thus having a high recall of 80% for decision dependencies. The personal loan eligibility example starts with *The following are the factors that affect your eligibility:* and the tool wrongly identified "factors" as a concept. In the Student AID US example there is a concept "school for specific program" which the tool identifies as two concepts "school for" and "specific program". We conclude that the DMN tool can be used for building a DRD since it can identify almost all of the concepts and requirements (high recall). Next, the DMN tool might label certain words too quickly as a concept (occasional low precision) and sometimes making the right link between concepts is also tricky for the tool. This matches with human behavior where these things are also a tricky aspect of decision modeling. Lastly, we noticed that on websites a simple bullet point structure, with each bullet point describing a dependency, occurs regularly. The decision tool can deal quite well and consistently with these. Currently, the only other approaches to extract DMN models from text are pattern-based approaches Arco et al. (2021), Etikala et al. (2020), Quishpi et al. (2021) as such these only understand patterns that have been provided. As of now, these approaches have only been tested on a limited number of patterns. The main advantage of pattern-based approaches is that these can be fully and easily explained. Even though, Section 6.2 specifically provides an explanation of how the deep learning models work, deep learning explanations are in general less transparent than pattern-based approaches. On the other hand, the proposed deep learning approach is more robust to new patterns as it also makes use of transfer learning from a big corpus of text. The other main advantage of the proposed deep learning pipeline is that it is language independent. If a similar tagged dataset in another language is provided to a correct BERT model than a similar performance can be expected. With pattern-based approaches one would have to analyze the possible patterns for each language again. A final clear disadvantage of the deep learning approach is the need to gather and label a dataset which is very time-consuming.

8.2. Limitations and future work

Currently, coreference resolution is not powerful enough to be fully automated and mainly struggles with vocabulary and synonyms which are crucial to identify words referring to the same concept. This problem is still being studied (Joshi, Levy, Weld, & Zettlemoyer, 2019; Sukthanker, Poria, Cambria, & Thirunavukarasu, 2020). Next, only sentences containing maximum two decision dependency levels are covered within the dataset. We believe though that it covers a wide array of realistic sentences. Moreover, we assume that sentences are grammatically correct and in English. This approach is feasible for other languages as it would require a similarly tagged dataset of sentences and a BERT model for that language, e.g., CamemBERT (Martin et al., 2019, French), GottBERT (Scheible et al., 2020, German).

As future work, extending the dataset with more real life sentences is planned to make both the classifiers and the extractors more robust for new patterns and cases. Even though the decision modeling tool contains a decision logic extractor, it still needs further research. The **irrelevant for decision modeling vs relevant for decision modeling** sentence classifier (Goossens et al., 2021a) will be evaluated on real life sentences and an extra label **process** identifying sentences describing a process will be added. We are also planning to perform a complete quantitative comparison with a more extended study of Etikala et al. (2020) in the future. Further additions to the tool such as a better feedback mechanism to update the concepts or manual classification of sentences are also planned.

9. Conclusion

In this paper, decision dependency extraction using deep learning has been implemented and evaluated on sentences coming from DMN application domains. We conclude that BERT is able to extract these decision dependencies. Moreover, a first evaluation of text classification for decision modeling has also been performed where BERT is also performing better than traditional machine learning models. For the first time, a tagged dataset for decision modeling is made available. Lastly, the first DMN model extraction tool from text using deep learning is also made available containing all main functionalities: text classification, decision dependency extraction and decision logic extraction. With this tool, it is now possible to extract decision dependencies and logic (semi)-automatically.

CRediT authorship contribution statement

Alexandre Goossens: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing, Visualization. **Johannes De Smedt:** Conceptualization, Validation, Writing – review & editing. **Jan Vanthienen:** Conceptualization, Validation, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data will be referenced with a Github link in the paper.

Acknowledgments

We would like to express our gratitude towards Michelle Claessens and Charlotte Parthoens for developing an early version of the decision model extraction tool. We would also like to thank the reviewers for their constructive feedback that resulted in a better version of this paper.

References

Abadeer, M. (2020). Assessment of distilbert performance on named entity recognition task for the detection of protected health information and medical concepts. In *Proceedings of the 3rd clinical natural language processing workshop* (pp. 158–167).

Adoma, A. F., Henry, N.-M., & Chen, W. (2020). Comparative analyses of Bert, Roberta, Distilbert, and Xlnet for text-based emotion recognition. In *2020 17th International computer conference on wavelet active media technology and information processing* (pp. 117–121). IEEE.

Alshemali, B., & Kalita, J. (2020). Improving the reliability of deep neural networks in NLP: A review. *Knowledge-Based Systems*, *191*, Article 105210.

de AR Goncalves, J. C., Santoro, F. M., & Baiao, F. A. (2009). Business process mining from group stories. In *2009 13th International conference on computer supported cooperative work in design* (pp. 161–166). IEEE.

Arco, L., Nápoles, G., Vanhoenshoven, F., Lara, A. L., Casas, G., & Vanhoof, K. (2021). Natural language techniques supporting decision modelers. *Data Mining and Knowledge Discovery*, *35*(1), 290–320.

Batoulis, K., & Weske, M. (2018). A tool for the uniqueification of DMN decision tables. In *BPM (Dissertation/Demos/Industry)* (pp. 116–119).

Bazhenova, E., Bülow, S., & Weske, M. (2016a). Discovering decision models from event logs. In *International conference on business information systems* (pp. 237–251). Springer.

Bazhenova, E., & Weske, M. (2016b). Deriving decision models from process models by enhanced decision mining. In *International conference on business process management* (pp. 444–457). Springer.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: Analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on machine learning* (pp. 160–167).

Danenas, P., Skersys, T., & Butleris, R. (2020). Natural language processing-enhanced extraction of SBVR business vocabularies and business rules from UML use case diagrams. *Data & Knowledge Engineering*, *128*, Article 101822.

De Smedt, J., Hasić, F., vanden Broucke, S. K., & Vanthienen, J. (2019). Holistic discovery of decision models from process execution data. *Knowledge-Based Systems*, *183*, Article 104866.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Dragoni, M., Villata, S., Rizzi, W., & Governatori, G. (2016). Combining NLP approaches for rule extraction from legal documents. In *1st Workshop on mining and reasoning with legal texts*. MIREL 2016, Sophia Antipolis, France: URL https://hal.archives-ouvertes.fr/hal-01572443.

Etikala, V., Van Veldhoven, Z., & Vanthienen, J. (2020). Text2dec: Extracting decision dependencies from natural language text for automated DMN decision modelling. In *International conference on business process management* (pp. 367–379). Springer.

Ezen-Can, A. (2020). A comparison of LSTM and BERT for small corpus. arXiv preprint arXiv:2009.05451.

Friedrich, F., Mendling, J., & Puhlmann, F. (2011). Process model generation from natural language text. In *International conference on advanced information systems engineering* (pp. 482–496). Springer.

Ghag, K. V., & Shah, K. (2015). Comparative analysis of effect of stopwords removal on sentiment classification. In *2015 International conference on computer, communication and control* IC4, (pp. 1–6). IEEE.

Goossens, A., Claessens, M., Parthoens, C., & Vanthienen, J. (2021a). Deep learning for the identification of decision modelling components from text. In *International joint conference on rules and reasoning* (pp. 158–171). Springer.

Goossens, A., Claessens, M., Parthoens, C., & Vanthienen, J. (2021b). Extracting decision dependencies and decision logic from text using deep learning techniques. In *International conference on business process management* (pp. 349–361). Springer.

Han, X., Hu, L., Mei, L., Dang, Y., Agarwal, S., Zhou, X., et al. (2020). A-BPS: Automatic business process discovery service using ordered neurons LSTM. In *2020 IEEE international conference on web services* (pp. 428–432). IEEE.

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.

Joshi, M., Levy, O., Weld, D. S., & Zettlemoyer, L. (2019). BERT for coreference resolution: Baselines and analysis. arXiv preprint arXiv:1908.09091.

Kjeldgaard, L., & Nielsen, L. (2021). NERDA. URL https://github.com/ebanalyse/NERDA. Accessed 3 December 2021.

Kluza, K., & Honkisz, K. (2016). From SBVR to BPMN and DMN models. Proposal of translation from rules to process and decision models. In *International conference on artificial intelligence and soft computing* (pp. 453–462). Springer.

Li, J., Sun, A., Han, J., & Li, C. (2020). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*.

Liang, H., Sun, X., Sun, Y., & Gao, Y. (2017). Text feature extraction based on deep learning: A review. *EURASIP Journal on Wireless Communications and Networking*, *2017*(1), 1–12.

Lin, D., & Pantel, P. (2001). Discovery of inference rules for question-answering. *Natural Language Engineering*, *7*(4), 343–360.

Lopez, M. M., & Kalita, J. (2017). Deep learning applied to NLP. arXiv preprint arXiv:1703.03091.

Martin, L., Muller, B., Suárez, P. J. O., Dupont, Y., Romary, L., de La Clergerie, É. V., et al. (2019). Camembert: A tasty french language model. arXiv preprint arXiv:1911.03894.

Munková, D., Munk, M., & Vozár, M. (2013). Influence of stop-words removal on sequence patterns identification within comparable corpora. In *International conference on ICT innovations* (pp. 67–76). Springer.

Ng, V., & Cardie, C. (2002). Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 104–111).

OMG (2015). OMG: Decision model and notation 1.0 (2015). URL https://www.omg.org/spec/DMN/1.0/ Accessed 8 January 2022.

Otter, D. W., Medina, J. R., & Kalita, J. K. (2020). A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(2), 604–624.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1532–1543).

Pradana, A. W., & Hayaty, M. (2019). The effect of stemming and removal of stopwords on the accuracy of sentiment analysis on indonesian-language texts. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 375–380.

Quishpi, L., Carmona, J., & Padró, L. (2021). Extracting decision models from textual descriptions of processes. In *International conference on business process management* (pp. 85–102). Springer.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), 9.

Ramshaw, L. A., & Marcus, M. P. (1999). Text chunking using transformation-based learning. In *Natural language processing using very large corpora* (pp. 157–176). Springer.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144).

Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence* (pp. 1044–1049).

Rosenstein, M. T., Marx, Z., Kaelbling, L. P., & Dietterich, T. G. (2005). To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning: vol. 898*, (pp. 1–4).

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Sansone, C., & Sperlí, G. (2021). Legal information retrieval systems: State-of-the-art and open issues. *Information Systems*, Article 101967.

Scheible, R., Thomczyk, F., Tippmann, P., Jaravine, V., & Boeker, M. (2020). GottBERT: A pure German language model. arXiv preprint arXiv:2012.02110.

Sinha, A., & Paradkar, A. (2010). Use cases to process specifications in business process modeling notation. In *2010 IEEE international conference on web services* (pp. 473–480). IEEE.

Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Machine Learning*, *34*(1), 233–272.

Sukthanker, R., Poria, S., Cambria, E., & Thirunavukarasu, R. (2020). Anaphora and coreference resolution: A review. *Information Fusion*, *59*, 139–162.

Torfi, A., Shirvani, R. A., Keneshloo, Y., Tavaf, N., & Fox, E. A. (2020). Natural language processing advancements by deep learning: A survey. arXiv preprint arXiv:2003.01200.

Valencia-Parra, Á., Parody, L., Varela-Vaca, Á. J., Caballero, I., & Gómez-López, M. T. (2021). DMN4DQ: When data quality meets DMN. *Decision Support Systems*, *141*, Article 113450.

Vanthienen, J. (2021). Decisions, advice and explanation: An overview and research agenda. *A Research Agenda for Knowledge Management and Analytics*.

Wang, Z., Dai, Z., Póczos, B., & Carbonell, J. (2019). Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11293–11302).

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

Wyner, A., & Peters, W. (2011). On rule extraction from regulations. In *Legal knowledge and information systems* (pp. 113–122). IOS Press.

Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. arXiv preprint arXiv:1702.01923.

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *Ieee Computational IntelligenCe Magazine*, *13*(3), 55–75.