POLITECNICO DI TORINO

MASTER's Degree in ICT for Smart Societies

MASTER's Degree Thesis

An ontological approach to exploit Natural Language Processing for blackout analysis in power systems

Supervisor

Prof. TAO HUANG

Candidate

MARTINA BAIOCCHI

July 2022

Summary

In recent years, the frequency of blackouts and the extension of their damage to power systems have been on a rise due to climate change, accidental and malicious incidences. However, few studies can be found analyzing the causes of the blackouts because the large amount of information over the Internet complicates the challenge of understanding the specific cause and keeping track of each outage. In addition, the lack of an ontology tailored for the blackout analysis is the proof and, at the same time, one of the causes for the missed chance of exploiting Named Entity Recognition (NER) and Relationship Extraction (RE). Therefore, in this study, a framework for the automatic collection and analysis of data about blackouts is proposed based on an ontological approach and Natural Language Processing (NLP), which extracts information from human-written text, allowing researchers to better focus on the entities that play a major role in power outages and the relations among them. More specifically, after defining the ontology suitable for blackout analysis, an NLP software in Python, *spaCy*, was selected, and documents were obtained from Google News. Based on the collected news articles, different experiments were then conducted to study the impact of the proposed ontology, the real core of this work, on the NLP model. The comparative analysis shows that by using the proposed ontology, even with a very small dataset, the results are promising. However, a larger dataset, especially if composed of technical reports, would be beneficial to use the ontology in practice, thus finally covering the gap between the exploitation of the NLP for blackout analysis compared to other fields. This way, in the next future, it would be possible to additionally rely on this technology for the monitoring of blackouts in power systems, building high-performing models that could be fine-tuned for more specific purposes, such as the creation of automatic databases or the study of threats that cause a specific effect on the electrical grid.

Acknowledgements

This work could have not been completed without the support of my family and friends. In particular, I owe a lot to my colleagues who enriched my university experience and helped me overcome some of my insecurities.

A special thanks also to Prof. Tao Huang who proposed such a challenging thesis, getting me out of my comfort zone and thought me a lot about the realization of scientific papers.

I dedicate this thesis to those who are no longer with us but also to my cousins, hoping that their life will be as full of surprises as mine has been, academically and otherwise.

Publication

The following project includes and expands the work presented at the IEEE MELECON 2022 conference and published in:

Tao Huang, Martina Baiocchi, and Xia Lei. «An ontological approach for automatic tracking causes of blackouts in power systems». In: *2022 IEEE Mediterranean Electrotechnical Conference (MELECON)*. 2022.

Table of Contents

List of Tables

List of Figures

Acronyms

AI
: Artificial Intelligence

BERT
: Bidirectional Encoder Representations from Transformers

GPT
: Generative Pre-trained Transformer

GPE
: Geopolitical Entity

LSTM
: Long Short Term Memory

ML
: Machine Learrnig

Mask LM
: Masked Language Modeling

NL
: Natural Language

NLP
: Natural Language Processing

NSP
: Next Sentence Prediction

RE

Relation Extraction

SG

Smart Grid

SVM

Support Vector Machine

UI

User Interface

Chapter 1

Introduction

Studies spanning from 2003 to 2019 [1, 2, 3] have shown that the frequency of blackouts has been increasing over the past few years. Relevant examples are the power outages that lately hit the USA. In August 2020, more than 13.8 million American customers were left in the dark after hurricane Isaias severely damaged the power system [4]. Not even a year later, in February 2021, severe winter storms caused a spread infrastructure failure in Texas, USA, with economic losses that amounted to more than $196 million [5]. In other parts of the world, the situation is not any better. In January 2021, supply was cut for around 200 million customers in Islamabad and other major cities in Pakistan after a technical fault caused a blackout [6]. The huge extension and negative effects of these outages are proof of the major role electricity plays in modern society as well as a warning against the scale of consequences we could be facing in the near future.

Therefore, studies on past blackouts would be beneficial for designing the adequate reinforcement of the electric grid, especially considering the large variety of possible threats. The large amount of required data and the difficulty in understanding such complex natural phenomena and electrical grids pose a challenge to researchers that requires new tools.

Until now, hundreds of reports needed to be manually scanned, as done by [1, 2, 3]. However, this task could be overwhelming for humans, time consuming, and it could leave room for errors. Resorting to Natural Language Processing (NLP), a branch of Artificial Intelligence (AI) on the rise in the last two decades, could be a solution. In this work, an ontology-based framework for blackout analysis is proposed, exploiting Named Entity Recognition (NER) and Relation Extraction (RE). These are two NLP tasks, aiming at detecting and classifying entities and the relations among them in a document. Through spaCy, a tool for designing NLP models, different study cases show promising preliminary results and the potential of the proposed framework.

The remainder of this thesis is organized as follows:

- **Chapter 2**: review of the start-of-art of natural language processing and relevant software;
- **Chapter 3**: introduction of the proposed framework for blackout analysis in power systems;
- **Chapter 4**: description of the proposed ontology and associated hierarchy of entities and relationships;
- **Chapter 5**: a brief description of transformers, the technology at the core of NLP in this project;
- **Chapter 6**: walkthrough on the main tasks of the framework, from data collection to the the creation of different NER and RE experiments;
- **Chapter 7**: performance comparison of the NLP experiments obtained in the previous step;
- **Chapter 8**: conclusions.

Chapter 2

State of Art

2.1 NLP and relative tool

The enormous amount of content on the Internet has still a potential that has not been completely exploited yet [7], largely due to the difficulties for humans to go through it. It is no surprise, then, how it is a breeding ground for Natural Language Processing (NLP) [8]. From companies trying to understand their customers' tastes through Sentiment Analysis, to translation among languages, to text-voice commands conversion and document summarization, this new AI branch seems to promise the automation of many language-related tasks, either full or partial.

In particular, the rise of NLP constitutes a new bridge between the human and the machine world when it comes to data analysis [9] from unstructured text. The possible exploitation of this new technology is however based on the manual creation of a corpus and an ontology, specifically designed for the desired field of use. Therefore, a preliminary study of the language is required and a huge collection of texts is fundamental for the practical implementation. Consequently, it is not surprising that many ontologies have been created only for theoretical purposes [10]. Different researchers tried to solve this problem by extracting conceptual models directly from documents. As much as this approach showed good results in the analysed cases, they included only simple types of Natural Language (NL) notation [11]. This lead to the exclusion of more complex cases, such as technical reports and news articles, which contain the most information about power outages, thus what is needed for the proposed framework. Moreover, it is logical to want to exploit them considering they should be in large quantity since, every time a blackout occurs, operators mandate their technicians and engineers to present a report and journalists write about the event. In particular, news articles are extremely easy to find, thus a good starting point for a corpus[1]. Albeit the reduction of time consumption is one of the main advantages of automation, reducing human errors when working with technical descriptions is also a good reason to opt for NER and RE, especially in a high expertise field such as power systems.

This brings us back to the problem at the root of NLP: creating an ontology. Designing an ontology is not as immediate as it could seem, since it must be both specific to the desired field and flexible, while remaining useful even when dealing with instances of semantic ambiguity [12]. Coupling these requirements with the need for a corpus of thousands of expressions could impede considering NLP in real practice, especially in those applications where the way is not paved yet for this new approach.

It is important to notice, however, that even when a large amount of training samples is not accessible, researchers are still testing the potential of NER with interesting findings in some specific fields, e.g. in finance [13]. This is why it would not sound absurd the idea of exploiting NLP to gather information on blackouts. Smart grids (SG) are embedded with many novelties by design, and new AI technologies could simply be the last addition, especially since they could immensely help with the large amount of data produced and needed to maintain such complex systems. Starting from a previous research on creating a systematic way for manual analysis of power outages reports [1], a new conceptual model of an ontology dedicated to blackout analysis is proposed to ease the application of NLP for this task, including aspects of these occurrences that were not previously considered.

The previous projects taken as examples focused more on NER. However, when it comes to understanding the evolution of complex phenomena, it is necessary to comprehend also how their composing elements interact with each other. This is why the proposed framework includes Relation Extraction. Exactly as for NER, in other fields, this NLP task has already shown good results. In 2018, a group of researchers trained least squares support vector machine (SVM) classifiers [14] with a medical corpus to extract interactions between proteins that were affected by mutations [15]. With the advancement of technology, extracting relations from biomedical texts was achieved with spaCy [16]. It is possible to notice how much this scientific field benefits from NLP also by simply looking at the numerous works present in the official "*spaCy Universe*" web page, where it is possible to share projects[2]. The clinical field also has taken an interest in NLP, as shown by the

[1] This aspect will be explained more in depth in Section 6.1.

[2] `https://spacy.io/universe`

existence of the spaCy-based package “scispaCy” [17].

Nowadays, spaCy is only one of the many NLP tools that are available, thus choosing one is not immediate. To avoid confusion, the selection of the NLP tool should focus on following the objective of the specific project. For instance, if looking only at performance, Stanford NLP would seem the best candidate, since in past studies it outperformed its competitors [18]. However, since the final goal of this study is to test a new ontology rather than designing a new NLP model ready for production[3], the opportunity to work with less setting up was given priority. This decision is also supported by the fact that proving the validity of this proposal automatically means future NLP models could be based on it, even built with more advanced software.

As a result, spaCy was selected. Not only this tool is now considered an industry standard to create NLP models, but its design aiming for production readiness and its efficiency on both GPU and CPU [19] guarantee to obtain results in a reasonable amount of time, even when data is scarce. This is also possible thanks to the auto-adjustment of internal hyperparameters based on the given dataset that is easily obtained with the official “quickstart widget”, which generates the starter setting configuration for each specific use case [20]. Another advantage offered by spaCy is the capacity of saving the best performing model during training by keeping track of the F-score, whichever epoch it was obtained from, thus avoiding the study for the optimal number of epochs. Finally, spaCy allows to design models based on transformers, a technology that will be described more in depth in Chapter 5.

As far as RE is concerned, although an alternative toolkit has already been created [21], the option not to rely on additional tools has been preferred. Considering simplicity a must in the proposed framework, basing both Named Entity Recognition and Relation Extraction only on spaCy was concluded to be a better solution. Moreover, as already mentioned for NER, once the framework has been proven to work, future projects could implement such task also with a different software. The major keys of the proposal, including its simplicity, can be found in Chapter 3.

[3]See Chapter 4 for more details.

2.2 Annotation tool and formats

Before discussing the framework of this project[4], it would be better to discuss the chosen annotation tool and formats to have a clear idea of the annotation process as a whole.

As for the NLP software, many tools exist to annotate text files. The choice that seems the easiest would be *Prodigy* [22] which would guarantee perfect integration with spaCy, since they were developed by the same company [23]. However, due to financial constraints, it was preferred to opt for a free software. This limited the options to *doccano*, *brat* and *INCEpTION*. Unfortunately, even though *doccano* [24] is known for its ease of use, it does not have the option to label relations, so it could not be selected. Between *bert* and *INCEpTION*, the first is more beginner-friendly, however the download link is no longer working [25] and the manual installation is discouraged by the many perquisites and passages that are needed that not only require time but could cause errors. This meant that *INCEpTION* [26] seemed a good compromise between the necessity to have a tool that included the possibility of relations labeling and an easier-to-follow setup. Its user interface (UI) is more complex than the UI of the other two options, however, the manual is very clear and once tested out on a small sample of text documents, the large variety of options of this software turned into an advantage rather than a drawback.

In particular, the possibility to save the annotations in many formats allowed a better ability in handling them to then translate them into the required spaCy format. Indeed, spaCy needs a specific JSON structure to train the models on Relationship Extraction, which is automatically obtained from *Prodigy* but not from third-party tools. However, after saving the annotations as UIMA CAS JSON, as allowed by *INCEpTION*, it was possible to store them in the required JSON file simply by running a custom script (Appendix A, more details in Section 6.1).

As far as the annotation of entities is concerned, spaCy requires a basic BIO format. Therefore, among the many options available from *INCEpTION*, CoNLL-2000 was selected. The other CoNLL versions were indeed more complex and included information that would have not been used, so it was preferred to go with the simplest.

[4]See Chapter 3.

Chapter 3

General Framework

The proposed framework (Fig. 3.1) is the very first trial of analysing blackouts with NLP. As such, it was appropriate and necessary to focus on the more practical aspects of this task, starting from the data collection to the production of final results and respective discussion:

1. *Ontology creation*: all the necessary entities and the relationships among them are created, considering several major aspects of blackouts (Sec.4);

2. *Data collection*: text documents (in this specific case, news articles) about power outages are automatically downloaded from Google News (Sec. 6.1);

3. *Data statistics*: statistics about the collected blackouts, now present in the dataset, are computed (Sec. 6.2);

4. *Splitting for the training and test datasets*: for NER, 80% of the downloaded text documents are used for the training dataset, while the remaining 20% for the test dataset; for RE, 80% for the training dataset, 10% for the evaluation set and 10% for the test dataset;

5. *Annotation*: The entities (for NER) and relations (for RE) in all the datasets were manually labeled;

6. *Training and testing*: for NER, different experiments (individualized by the number of considered entities and articles) were trained to test various factors; for RE, test cases were obtained considering firstly the most four present relations and then only the most frequent relation in the dataset, also modifying the value for the "max_distance" parameter (Sec. 7.1);

7. *Results and discussion*: the obtained results for the different cases were compared and discussed (Sec. 7).

The new ontology is the core of the framework, innovating the one typically used to create models based on Machine Learning (ML), without overturning it completely. As a further benefit, it is possible to run some of the tasks in parallel (e.g. “Ontology creation” and “Data collection”), which makes this framework suitable for research teams from different backgrounds to work collectively.

As the software is concerned, it is possible to see how all the tasks can be obtained with any preferred tool, since no block is specific to spaCy (the choice for this particular project). Moreover, this framework can be expanded to reach a different goal (e.g. not stopping at the result discussion, but adding visualization) or embedded in a bigger one (e.g. building an application where analysing blackouts is a mean to reach the final objective).

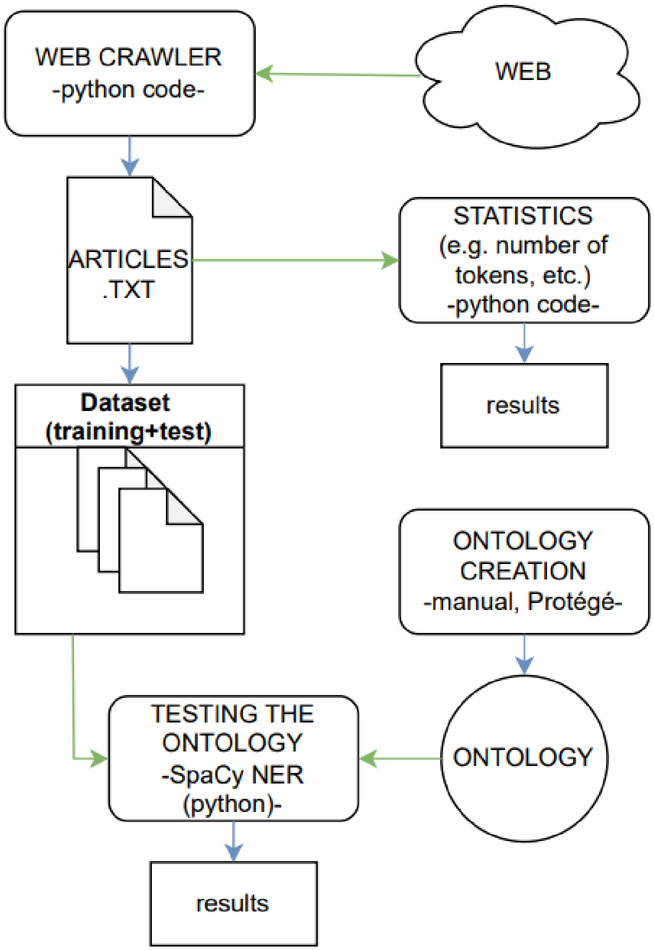


Figure 3.1: Framework of the proposed ontological approach.

As explained, creating the ontology is the core of the proposal and, if flexibility was considered a requirement, simplicity was almost an obligation. Unfortunately, in Chapter 2, one of the main questions was about the lack of exploitation of NLP in smart grids, which seemed a contradiction both to the nature of smart grids themselves and to the promising results obtained in other fields. Even though it is not possible to speak for an entire engineering community, the large amount of research about Machine Learning (ML) for power systems[1] suggests an inclination to focus only on data when coming from smart meters [28] and/or other sensors. The idea of exploiting human-generated data might seem behind the times, especially considering the short interval (down to 15 minutes, in some cases) and the high reliability of data from smart meters [29]. However, at least nowadays, although widely used algorithms based on them can help with load/generation balance, there is still some information missing from what sensors can tell: the connection among causes and effects that lead to a blackout and the possible solutions. Those are prerogatives of the experts, whose years in the job have given them the necessary experience to understand how different phenomena may or may not be a threat and, in case, how to address weaknesses in power systems. On a less precise level, also journalists can explain to the common public how and why the grid failed, which is why for this project it was possible to resort to news articles as text documents. A more detailed explanation of the reason for this choice and the process of data collection will be provided in Section 6.1.

[1]The topics of conferences could be taken as example [27].

Chapter 4

Ontology Creation

Engineering technologies and procedures are always based on standardization. This avoids errors [30] and allows works to still be functioning through the years and geographic locations. Without standardization, the inter-operation among systems designed by different teams, the expansion of already existing ones and their maintenance would not be possible. It is not surprising, then, that when it comes to NLP, having a well-defined structure that describes the field of use at hand is fundamental. The schematization of reality has been a topic of discussion for years, firstly present in philosophy and recently in engineering, where it can be summarized as "explicit specifications of conceptualizations" [31]. Since computers cannot understand real objects and concepts if not through numbers, it is necessary to translate a field into a scheme before building NLP models able to analyze it. Moreover, the scheme is needed by human annotators in order to minimize errors (due to wrong personal interpretation) and obtain uniform labelling of the documents, regardless of when and where such task is performed. In particular, the scheme is based on two basic components: entities and relations.

Entities describe the basic concepts of a certain field [32], representing real objects that can be either abstract or physical. When designing an ontology, listing them would not be enough. They have a precise place in the structure, considering all the possible relationships that link them. Moreover, they need to be well-defined to facilitate the annotators and avoid possible confusion. A too-strict definition, however, might cause a lack of flexibility and compromise the inclusion of all the possible declinations of the concept they represent. Compensating this by adding a huge amount of too specific entities would not be a good solution since, in practice, the first approach to training a new NLP model should be based on broad definitions [33]. Moreover, the resulting ontology would be overly complicated for humans, too. Considering spaCy already provides some models trained on a set of entities for generic purposes NER[1], it was seen appropriate for the proposed ontology to incorporate some of them while introducing new ones specific for blackout analysis. (Fig. 4.1).

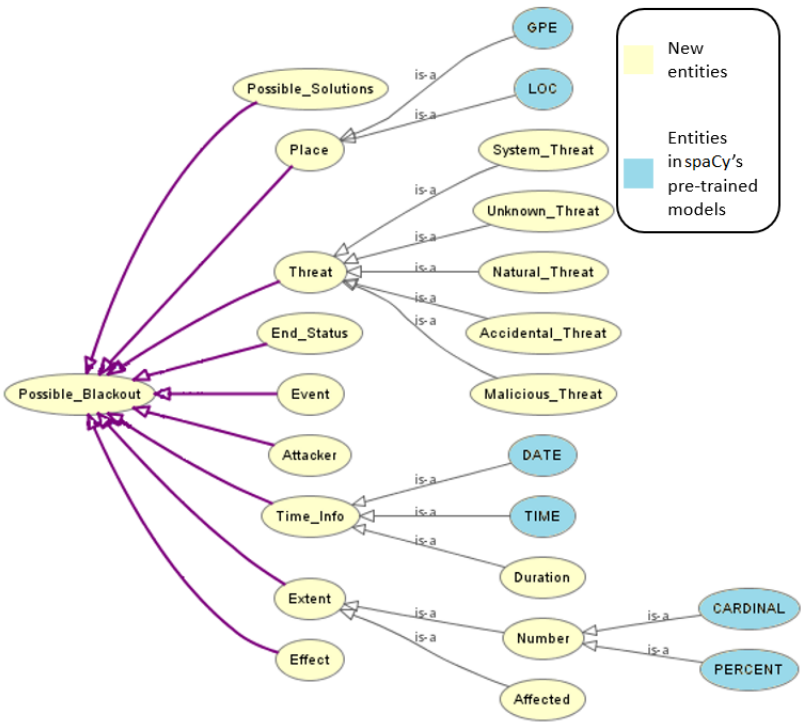


Figure 4.1: The hierarchy of entities. In capital letters, those already present in spaCy pre-trained models.

Taking into account this is the first ontology designed to analyze blackouts, the result covers many aspects of power outages, while still leaving room for future improvements, such as the addition of new entities or the rearrangement of presented ones. This can be useful when dealing with projects that focus on some particular cases or if the advancement of technology will require new entities. Take the "Threat" entity as an example: its sub-classes include all the major possible dangers to power grids presently; in the future, a new emerging threat could easily be added to the ontology without changing the whole structure. Also, considering the rise of cyberwarfare against electrical systems, new entities could be created to describe the different types of attacks adopted by hackers.

[1]To easily recognize them, they are always written in capital letters.

Now that the required characteristics for a good ontology have been explained, there are obviously some aspects that are specific to the proposal. Before deciding which entities should have been included, it was imperative to set a clear objective: the extraction of information to better understand how and why power outages occur. To reach this goal, many approaches could have been taken but, as stated, simplicity and understandability had to be ensured, thus answering simple yet precise questions seemed the best way of proceeding. This resulted in selecting a criterion to find the macro-classes of the ontology, which is answering one of the so-called five "W's": *What*, *When*, *Where*, *Who*, *Why* (Table 4.1).

Table 4.1: Macro-classes and respective question.

Macro-class	"W"	Question
End Status	*What*	What was the final state?
Possible Solutions	*What*	What could be done to prevent it?
Place	*Where*	Where did it occur?
Extent	*Where*	Where could it be drawn on a map?
Time Info	*When*	When did it happen?
Attacker	*Who*	Who is responsible for the attack?
Threat	*Why*	Why did the blackout occur?
Event	*Why*	Why did the threat(s) become a problem?
Effect	*Why*	Why did the system react poorly?

To comprehend something as complex as a blackout, entities are not enough. Relationships among them can help to retrace the chain of events leading to a power outage (Table 4.3). This means not only NER could be exploited but also Relationship Extraction (RE) (Fig. 4.2). However, the selection criterion for relations needed to be different. For those linking entities on the same level, it was possible to follow a cause-effect approach. An example could be a threat that makes some tree branches fall on electric wires (a *Threat* that causes an *Event*). For others, it was evident a passage from a general case to a more specific one. For instance, if the day of the blackout is known, in the text sometimes also the exact hour is present (from a *DATE* to a *TIME*). Finally, similarly to entities, it is also possible to add new relationships or modify already existing ones.

As already explained, an ontology also needs to be very clear for annotators; therefore, short descriptions of the classes and the relations have been provided, as well as some simple examples for the entities, which can be read in Table 4.2. As far as testing an ontology is concerned, there is not a method experts agree upon [34]. Moreover, the test often consists in comparing multiple ontologies designed for the same field. In this project, however, a completely new one is proposed, for which there was not even a corpus yet, so it has been decided to follow a variation of what Brank et al. defined as an “application-driven evaluation”.

Table 4.2: Identified main entities for blackout analysis.

Entity	**Description**
Threat	Condition damaging the system.
Threat:Malicious_Threat	Attack by human action.
Threat:Natural_Threat	Natural phenomena.
Threat:Accidental_Threat	Errors or unexpected phenomena.
Threat:System_Threat	Conditions intrinsic to the system.
Threat:Unknown_Threat	Unknown phenomena/actions.
Place	Geographical location.
Place:GPE	Geopolitical Entity.
Place:LOC	Locality.
Time_Info	Time-related information.
Time_Info:DATE	Date of the blackout.
Time_Info:TIME	Time (hour/part of the day).
Time_Info:Duration	Duration of the power outage.
Extent	Magnitude of the blackout.
Extent:Affected	Affected entities by the blackout.
Extent:Number	Number of affected entities.
Extent:Number:PERCENT	Percentage.
Extent:Number:CARDINAL	Cardinal number.
Event	Physical event caused by the threat.
Effect	Effect on the system.
Attacker	Entity behind unknown/malicious threats.
End_Status	Occurrence or not of a power outage.
Possible_Solutions	Suggested solutions against failures.

Table 4.3: Relationship extraction among proposed entities.

Relationship	**From**	**To**
_at_	DATE	TIME
_by_	Malicious_Threat	Attacker
_in_	GPE	LOC
causes	Threat, Event	Event
ends_in	Threat, Event, Effect	End_Status
number_of	Number	Affected
results_in	Event, Effect	Effect
solved_with	End_Status	Possible_Solutions

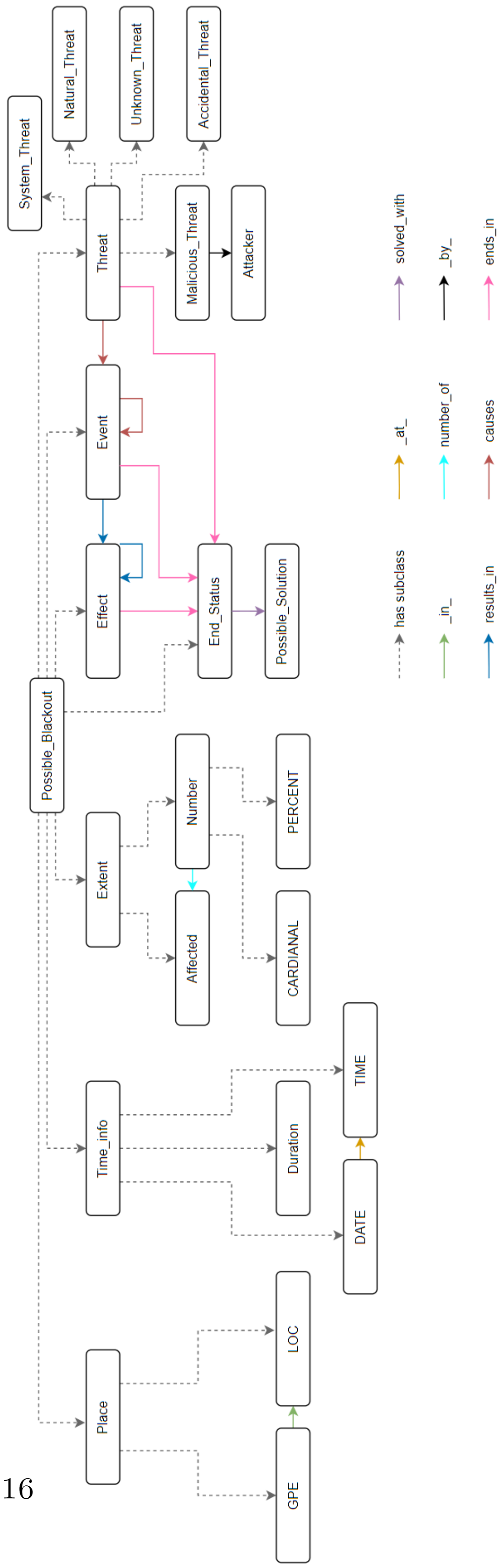


Figure 4.2: Entities and their relations in the proposed ontology (implemented in Protégé [35]).

Chapter 5

An overview on transformers

Until 2017, translation was a complex task assigned to Long Short Term Memory (LSTM) neural networks, although their drawbacks were not negligible. Since words needed to be passed one at a time and multiple steps were required for learning, they were slow. Additionally, LSTMs also lack true bidirectionality, so they were not entirely able to capture the meaning of words [36]. Despite their name, even the so-called “Bidirectional LSTM” networks do not completely overcome these problems, because they actually simply learn the context left-to-right and right-to-left to then concatenate the result, missing some information from the original one [37].

A new technology, however, shortly would have changed the approach to translation and NLP in general. Vaswani et al. designed an architecture (Fig. 5.1), called “Transformer”, that could address the shortcomings of LSTMs by being truly bidirectional. This meant it could exploit parallelization to better learn the context and be trained faster [38]. It is composed of two main parts, repeated multiple times: an encoder and a decoder. Another advantage over the complex structure of LSTM networks is the clear division of tasks assigned to each one of them. To give an example regarding translation, the encoder could be seen in charge of understanding the context from the text in the original language, while the decoder learns the relation between the original language and the one the text needs to be translated to. Along with this strong point, it is also possible to see how a further advantage is that both the components have some understanding of language.

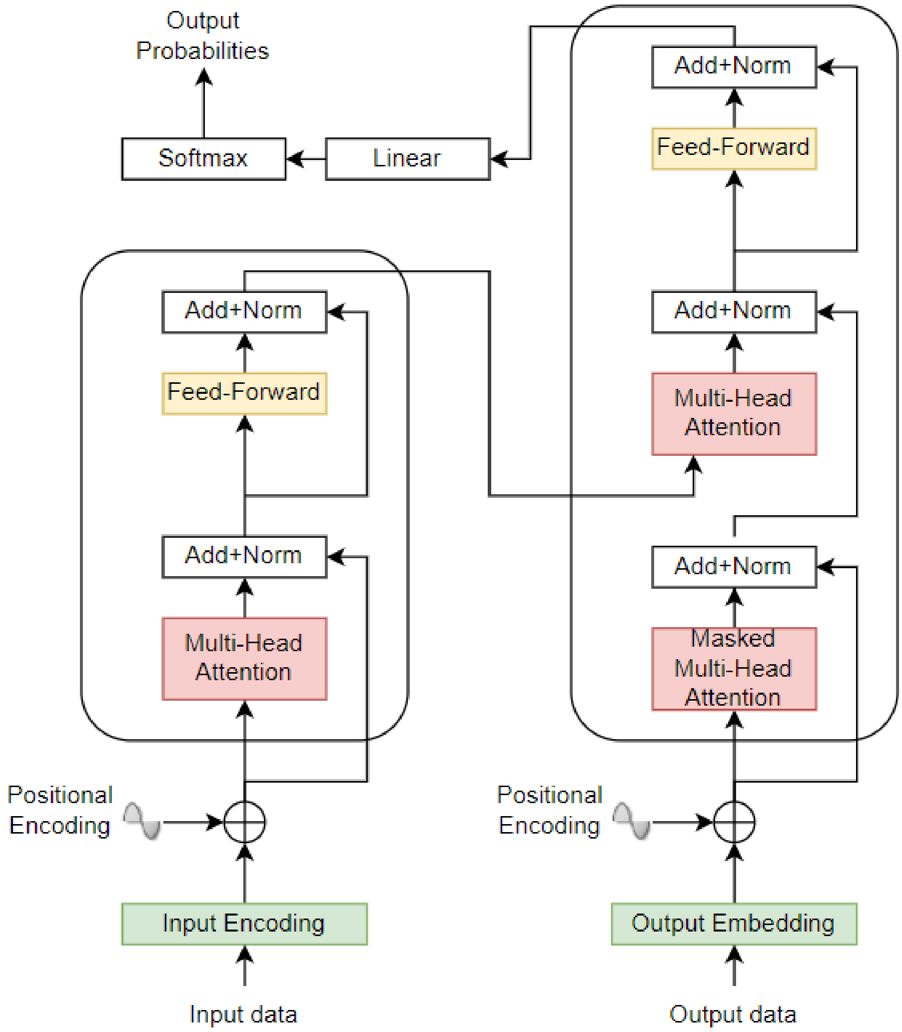


Figure 5.1: The architecture of the original Transformer. Rielaboration of an image from [38] by M. Baiocchi.

For this reason, other architectures stemmed from the first proposal, such as the Generative Pre-trained Transformer (GPT) which is the result of stacking decoders, and Bidirectional Encoder Representations from Transformers (BERT) obtained by stacking encoders (Fig. 5.2). BERT in particular is at the core of the models that were built to prove the proposed ontology can be used in practice and will be the object of a more detailed description. Nevertheless, a brief explanation also on how decoders work will be provided to have a clearer picture of the general functioning of transformers.

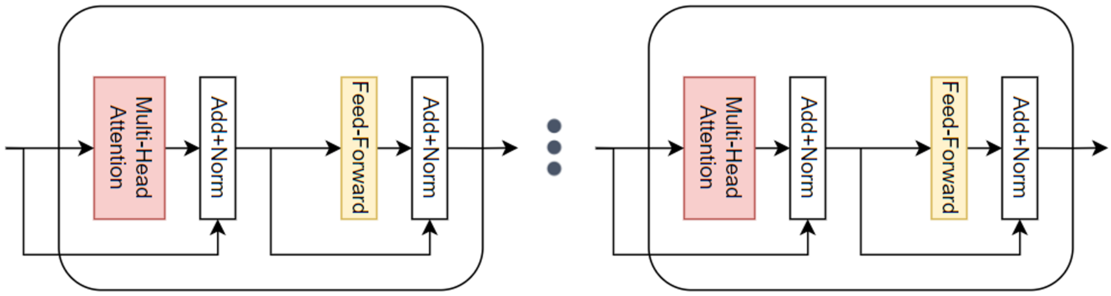


Figure 5.2: The architecture of BERT. Rielaboration of part of an image from [38] by M. Baiocchi.

5.1 Encoder

The first task required of an encoder is to translate words into embedding vectors. Sometimes, however, a word can have different meanings, so always associating it to the same vector representation would result in a loss of context. This is why encoders in transformers not only convert words into vectors but then perform positional encoding on them (Fig. 5.3). In practice, this means applying a non-linear function that takes into account the position of the word in the sentence.

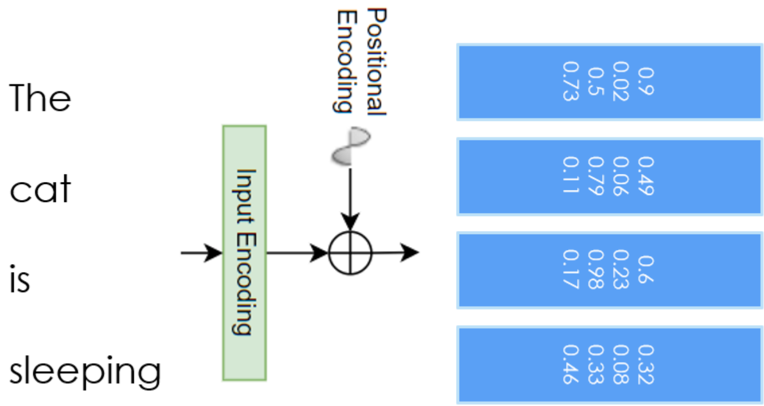


Figure 5.3: Positional Encoding in the BERT architecture.

After positional encoding, it is necessary to understand which are the most important parts of the sentence. To achieve this goal, the vectors previously obtained enter the Encoder Block and pass through the Multi-Head Attention Layer. Here, for each word, the relation to the others is computed and stored in a so-called Attention Vector. However, it would be not enough to compute only one per word or there could be the risk that the element representing the relation to itself would weigh way more than those representing the relations to the others. To solve this problem, multiple vectors are computed for each part of the sentence and then concatenated to form larger Attention Vectors, one for each word (Fig. 5.4). To compute a single larger Attention Vector, we have in parallel triplets of vectors to represent different components of a word and a softmax function is applied to them. This operation is the Scaled Dot-Product Attention. After performing multiple operations of this kind, the respective results (i.e. the different attention vectors for a word) are then concatenated.

Before exiting the Encoder Block, the Attention Vectors are finally translated by Feed-Forward Networks into a more digestible form for the next component, either an encoder or decoder depending on the chosen architecture. This is where it is possible to exploit one of the advantages that transformers have over LSTMs: since

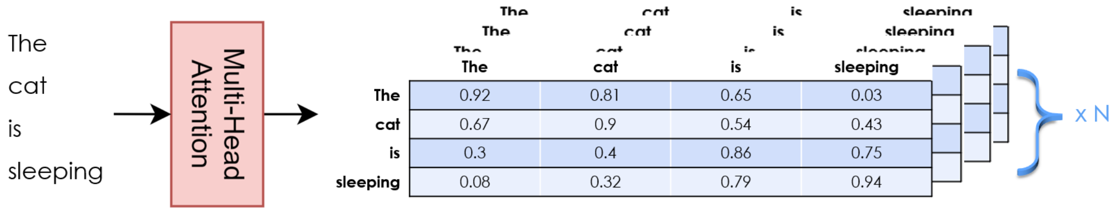


Figure 5.4: Multi-Heat Attention Layer in the BERT architecture.

the Attention layers are independent, by having multiple Feed-Forward Networks, parallelization can be performed.

As a note, after every layer, normalization is applied. A possible one that could be chosen is batch normalization, to help smooth the loss surface for a better optimization and the possibility to use larger learning rates.

5.2 Decoder

Decoders have all the components in common with encoders with the exception of a Masked Multi-Head Attention Layer, which is positioned between the Positional Encoder and the Multi-Head Attention Layer. A brief explanation of how it operates will be given, taking as an example translation from one language to another. In this layer, the Attention Vectors for the terms in the final language are computed taking into account the Attention Vectors for the original words obtained from the encoder and the previous words in the translated sentence. Practically, the next words are "masked", hence the name. If masking is not performed, there would be no learning.

At this point, the new objective is to understand the relations between these two sets of vectors to perform a mapping between the two languages. This is the task of the Multi-Head Attention Layer, which computes the vectors to represent the relation of each word both to the other in the same language and those in the different language and passes them to the Feed-Forward Layer.

As for the encoder, after each layer, normalization is performed. Finally, after the last decoder, the obtained vectors pass through a linear Feed-Forward Layer to have the number of dimensions equal to the words in the translated sentence and then to a Softmax Layer to obtain the probability distribution[1].

[1] For more information, please visit [39].

5.3 BERT: pre-training and fine-tuning

Even though BERT was originally designed for translation, nowadays it is used for many different NLP tasks (e.g. Sentiment Analysis). To achieve this variety of objectives, BERT needs first to be pre-trained in order to understand language and then it can be fine-tuned to solve a specific problem. In this project, already pre-trained BERT has been exploited. However, just as for the decoder, also a brief explanation on pre-training will be given, simply for clarity.

With this first task, BERT learns to understand context by performing simultaneously Masked Language Modeling (Mask LM) and Next Sentence Prediction (NSP) [40]. The input consists of two sentences, where some words have been masked. The goal of Mask LM is to output the sentences completed with the missing words, while the task of NSP is to understand if the first sentence is followed by the second (the output will be *1* if it does, *0* otherwise).

Before any operation can be performed, the tokens (i.e. words) need to be translated by some pre-trained encoders into embedding vectors. Each of them is obtained from three vectors: the first one is a pre-trained embedding, while the second and the third are respectively the encoding of the number of the sentence and of the position of the word in the sentence. These last two are needed not to lose the order of tokens, since they are passed to BERT in parallel. The output of Mask LM is also composed of parallel tokens which have the same size. They are then passed to a Softmax Layer to obtain one distribution each. Finally, the distributions obtained for the originally masked tokens are compared to the respective labels, which are represented by one-hot encoding, to compute the cross-entropy loss.

Once pre-trained is complete, it is possible to fine-tune BERT by replacing the final fully-connected layer with different output layers[2], depending on the chosen objective, and performing supervised training. This means that only the output parameters have to be learned, which makes the training fast. The input and the expected output depend obviously on the desired goal.

[2]e.g those designed by the developers of spaCy [41].

Chapter 6

Case Study

In this chapter, the process from data collection to the design of NLP models for this project will be retraced. Challenges and choices will be explained in order to clarify the flow of the entire work.

6.1 Data Collection

The starting point for any project is obviously collecting the necessary data. Originally, technical reports on power outages were considered a valid choice since they are more complete and written in the accurate scientific language used by experts. Unfortunately, it was not possible to rely on them for different reasons. The first is the lack of an open-source database containing a large amount of them, which would have been extremely helpful. Obviously, these documents are written by technicians and engineers working for the company whose system is affected by the blackout, so they are not divulged. Asking for permission to access them would not only be extremely time-consuming but it would often result in a negative answer.

The second option was to exploit the PDF files of some old reports that had been scanned for a previous work [1]. Different tools were tried to extract text from them but with unsatisfactory results. Both Python module "PDFMiner" and "PDF.co" API [42] failed due to multiple problems. Firstly, the majority of those files presented text that was not only divided into two columns, but the pattern was usually not constant due to tables and figures. Those were also another obstacle since their positioning could not be predicted and this meant that the inclusion of the text in the captions and in the tables was not avoidable a priori, but required to be manually removed. In addition, many documents presented lines at the beginning and end of pages, such as the copyright mark or the title of the current chapter, that would have needed to be manually deleted, too. Finally, one of the largest reports was so old that the ink was smudged and thus some words could not be extracted. Considering the human resources and time constraints of the project, the most logical solution was to find another type of data, especially taking into account that NLP models require thousands of samples to be trained, as explained by Montani, one of the developers for spaCy [33].

The best option seemed to be automatically downloading articles from the Internet. To achieve this goal, a web scraper was exploited in combination with the "GoogleNews" Python module to retrieve the articles and the entities and relations in them were manually labeled, using the Technical University of Darmstadt's INCEpTION environment. For entities, labelling followed the BIO-style and it was saved in CoNLL-2000 formatted files that were later translated in spaCy's binary files with a terminal command provided by the NLP tool itself. For relations, the labelling followed the UIMA CAS JSON format, which was translated into spaCy's binary files after being converted into JSON files with the spaCy's required pattern thanks to the combination of the code provided by Amamou [43] and a custom Python script (Appendix A).

It is obvious that all these procedures were very time-consuming; therefore, the number of collected news reports was limited to 120. The choice was based on the following criterion: only the first 90 articles about blackouts caused by natural phenomena and the first 30 due to cyberattacks were downloaded. This amount of documents was chosen considering the approach followed by Jabbari, et al. [13], who had to face similar challenges in their study.

6.2 Statistics of the dataset

Before designing any NLP model, an analysis of the downloaded news article was deemed beneficial. Information about different aspects of both the power outages and the language was extracted and the findings were discussed.

6.2.1 Power outages

Considering the focus of the project is to build an ontology (and a framework around it) to better understand blackouts, it was necessary to know which power outages were present in the dataset after being downloaded as explained in Section 6.1.

It is important to underline that the only condition on Google News was the language, i.e. English. Therefore, it is not surprising that the majority (91.5%) of the selected reports described blackouts occurred in North America, especially in the USA, or other English-speaking countries (e.g. UK and Australia). This is true in particular for those failures that were caused by natural threats. For malicious threats only five blackouts were found, one of which from the United States, while the others both from Asia and Europe. This is probably caused by a lack of episodes for this scenario, which means that the language condition does not heavily influence the first Google results. The weight of the absence of many examples of cyberattacks on grids is proved also from the temporal range the resulting news articles were written in. Specifically, the samples present in the dataset regard the following electric system failures:

- Ukraine (December 2015);
- India (March 2021);
- Iran (April 2021);
- USA (May 2021);
- Puertorico (June 2021);

Since there is no explicit date condition on the search, it was expected only the download of recent events. The presence in the dataset of the blackout caused in Ukraine by a Russian malware in 2015 is due to the low number of similar events. It should be concerning, however, how even all the other episodes are from last year (2021), showing cyberwarfare is expanding to the energy field. For the blackouts that were triggered by natural threats, on the other hand, the abundance of examples led to only downloading only those occurred in the last couple of years (2020-2022).

As for the monthly trend, for malware-caused episodes, there is not a specific period of the year they happened. For those linked to natural phenomena, on contrary and as expected, it is visible a spike in winter and summer. Indeed, storms, winds, and ice might damage important structures in the grid during the coldest months. In summer, the same might happen due to extreme heat both directly and indirectly, i.e. when the abnormally high use of air conditioning causes an excessive load and the demand cannot be met. However, if the mild climate justifies the almost complete absence of blackouts during spring, their rise in autumn was surprising (Fig. 6.1). The "culprit" is climate change, bringing abnormal temperatures and precipitation and, in particular, the USA experienced the damage "La Niña" left behind in the last trimester of 2021 [44].

Finally, an explanation of the choice of the search conditions needs to be given. It might seem too loose to have just one requirement (i.e. news article in English only), but it is important to remember that there were no previous studies about which blackouts would be better to consider. This means that adding temporal/spatial constraints could have led to the introduction of personal bias, which is obviously to be avoided. On the other hand, following this choice meant collecting data only on the most significant blackouts in terms of severity and socio-economic impact, which was viewed as a good approach to the problem. Moreover, this project started without the objective to study one specific kind of grid failure, as this work is the first of its kind. As long as its feasibility as base for practical applications is proved, the construction of datasets that depend on the case of use can be left to future studies.

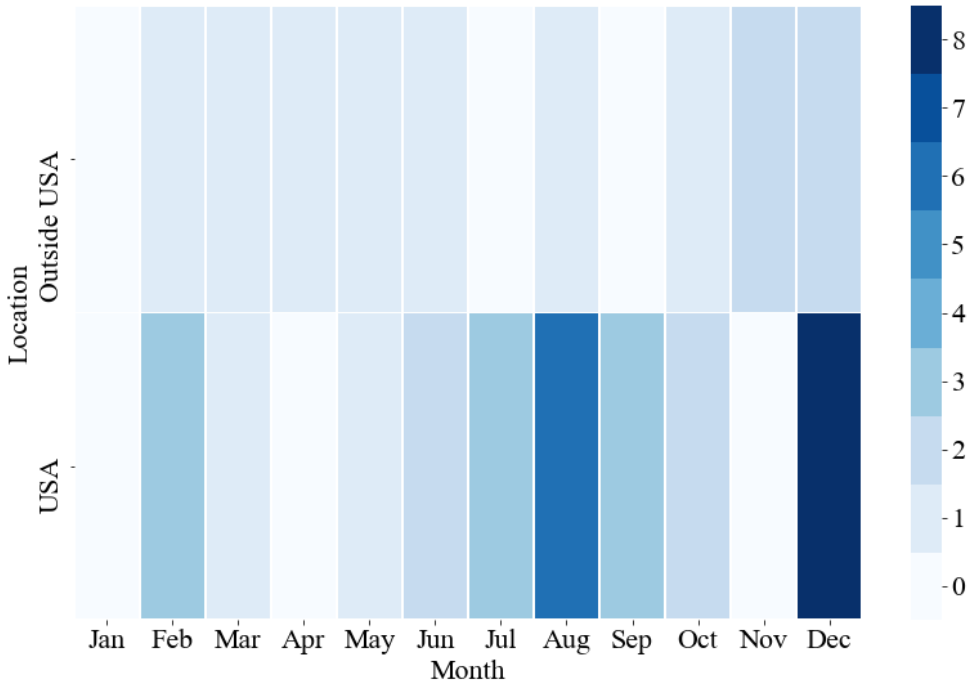


Figure 6.1: Number of power outages in the examined news articles by month and location.

6.2.2 Language

Considering that the entire project is based on NLP, a preliminary analysis of the selected blackouts was not sufficient. It was necessary also to retrieve information on the language used and on the expressions that constitute the dataset to train the NLP models.

As explained in Section 6.1, it was not possible to rely on technical reports, so the dataset is composed of news articles. This meant working with the journalistic language, which has the goal to be descriptive while still easy to understand also for non-experts. Moreover, some of the articles described the same blackout. This seemed to be a challenge, but it turned out to be an advantage. Even when talking about the same event, journalists did not use the same expressions but chose synonyms. This helped increase both the number and the variety of samples in the dataset, despite the low total amount of included news articles (Fig. 6.2). Indeed, looking at the whole pool of samples, a total of 230 expressions is found. However, when considering also repetitions and plural forms, training and test datasets combined include 1317 expressions and, since some of them are composed of multiple terms, the total number of words amounts to 2270. Even though this is obviously far from the amount of data required for an NLP model to be used in practice [33], but they are at least enough to test the proposed ontology, as it will be shown by multiple comparisons in Subsection 7.2. This also proves that, for the set goal, the chosen approach (i.e. not adding strict search conditions to avoid possibly introducing bias[1]) still resulted in a usable dataset.

For relations, unfortunately, the samples were very limited. This is why only the four most present were selected fot the project. Even so, the data scarcity was still an evident problem. Indeed, among all the 120 news articles, the following examples were found:

- *number_of*: 248,
- *ends_in*: 225,
- *_in_*: 159,
- *_at_*: 97.

Looking only at natural threats in the news articles, “storm” and “storms” are among the most frequent expressions, while the least used are those more detailed with adjectives or including the name of a particular storm (e.g. “Hurricane Gloria”). On the other hand, “cyberattack” and “malware” are the most preferred words for cyberwarfare under malicious threats. The reason is probably that during the last years, malicious attacks in physical forms, such as bombs, were less popular while attacks by hackers seem to be on the rise (as already noticed in Subsection 6.2.1). The stylistic choice might also derive from the necessary simplification of

[1] See Subsection 6.2.1.

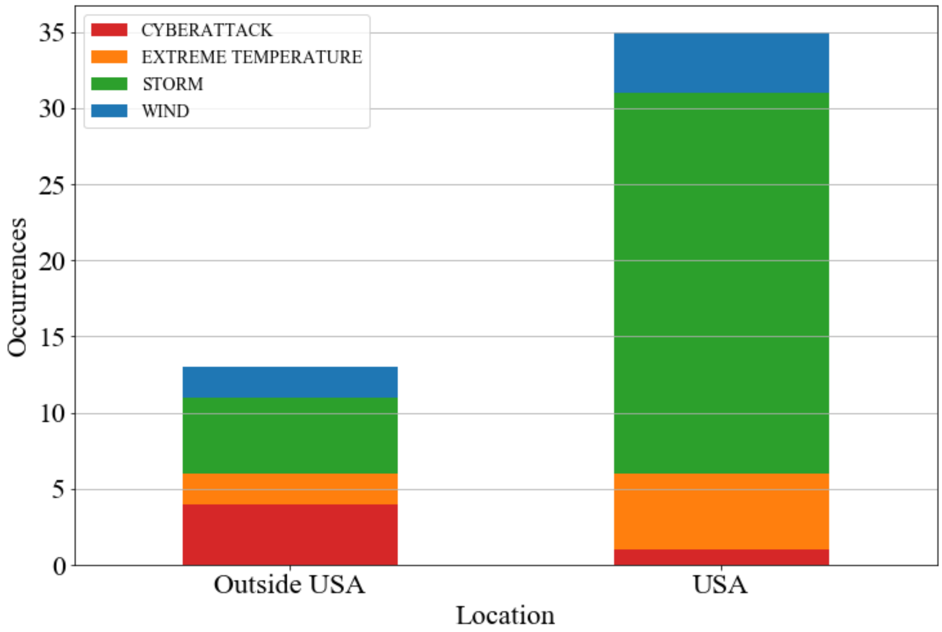


Figure 6.2: Main threats in the examined news articles. In "Storms" all the different types of storms are included.

such topics in journalism and the lack of variety of options to describe cyberwarfare (Fig. 6.3).

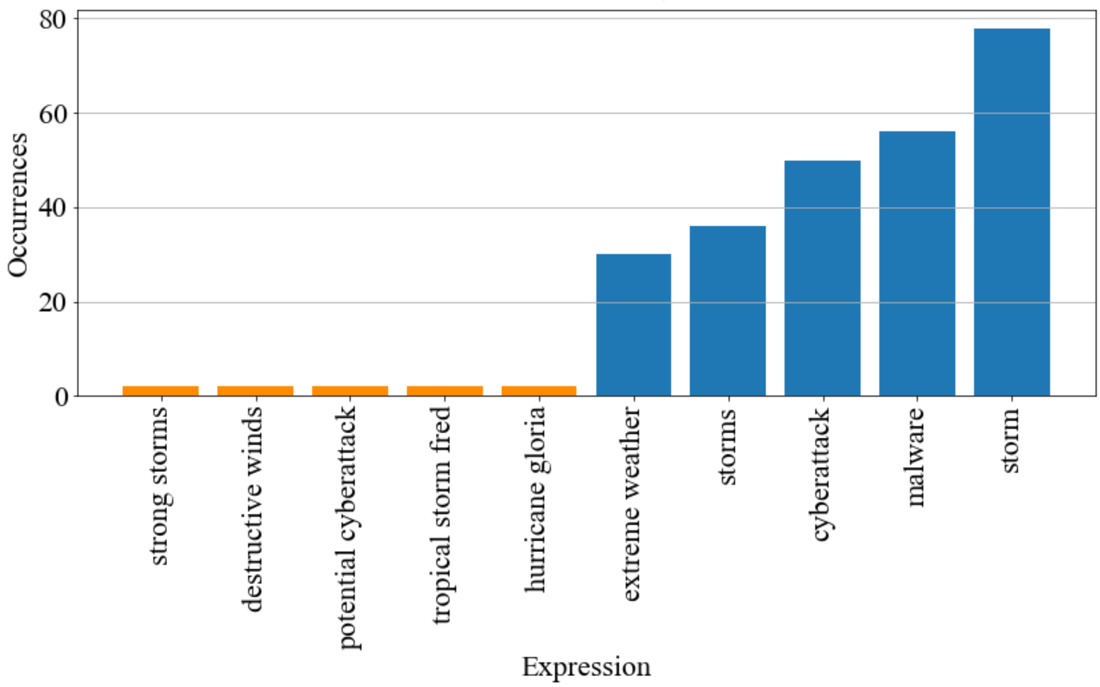


Figure 6.3: The five most and least frequent words used to describe threats in the examined news articles.

Chapter 7

Models and results

Once the dataset was completed, it was necessary to design the NLP models to test. Two approaches were possible: studying the hyperparameters optimization or the effect of different versions of the dataset and considered parts of the ontology on the performance. Considering that the objective of this work is proving the validity of the proposed ontology and the time constraints, studying the interaction among the over thirty parameters for each model did not seem the wisest choice.

7.1 NLP models

7.1.1 NER test cases

Before completing the 120 news article dataset, a previous version was composed of 84 articles, 30% about blackouts due to malicious threats and 70% due to natural threats. This meant it could be used to better test the impact of data scarcity on the model, compared to using all the documents. Another major factor that needed to be studied was the effect of the considered portion of the ontology on the performances.

Such ideas lead to the design of a first test case, *basic84*, whose objective was to detect and label only three entities: *Natural Threat*, *Malicious Threat*, and *End Status*. Among all the entities these were deemed essential as they inform on the cause and the final effect of a blackout on the electrical grid.

Remembering that spaCy comes with NER models already trained on some generic purpose entities, it was possible to test if this feature could be useful as a starting point also when dealing with completely new entities, thus a pre-trained model, *v0*, was built and re-trained with the 84 articles.

Considering the impact of including a larger portion of the ontology, the types of entities to be detected were increased. In *v1*, alongside the three previously present in *basic84* and *v0*, *Event*, *GPE*, *DATE*, *AFFECTED*, *CARDINAL*, and *PERCENT* were added. In *v2*, all the entities in the ontology were considered except those whose examples in the dataset were too few or completely absent (*Attacker*, *Accidental Threat*, *Duration*, *Effect*, *Possible Solutions*). As it is possible to notice, in these last two test cases, entities that are also present in the spaCy pre-training[1] needed to be found so, instead of starting from a blank model, a pre-trained one was used as base.

After the conclusion of those experiments, the effect of increasing the available samples for training and testing was analyzed. A new version, called *basic120*, was trained with the complete dataset with the same objective as for *basic84*.

At this point, it should not be surprising that the difficulty in finding data for the creation of an ontology is quite common. The number of total articles, as mentioned in Chapter 1, was chosen after the experiments of Jabbari et al. [13], who had to deal with data scarcity and also proposed to mitigate such problem by training the test case only for a few of their custom entities, while labeling all the remaining under a “miscellaneous” category. To test the validity of such approach also for blackout analysis, a new test case, *misc0*, was created considering the entities for the *basic120*, while all the unused were grouped under a new one, called “Other”. Taking into account the results obtained for *v0* (described in Subsection 7.2.1) and that *misc0* does not include pre-trained labels, this last test case was based on a blank model.

Finally, a test case, *NER_simplified*, was conducted to study if a higher samples-text ratio could have a positive influence on the performance. Such ratio was increased with the creation of a new dataset by selecting from the 120 news articles only the sentences containing *Affected*, *CARDINAL*, and *PERCENT*[2]. The reason for this choice is that this dataset was originally obtained to run two experiments concerning RE. However, since entities still need to be labelled for RE training, it seemed convenient to exploit it also to further experiment with NER. In this case, a pre-trained model was used, having only one out of the three entities from the new ontology (*Affected*). In Table 7.1, a summary of all the NER test cases is present.

[1]When they are explicitly mentioned, they are written in capitol letters.

[2]Entities in capital letters are those spaCy provides pre-trained models for.

Table 7.1: Construction of test cases.

Name	Number of news articles	Entities
basic	84, 120	Natural_Threat; Malicious_Threat; End_Status
v0	84	
misc0	120	Natural_Threat; Malicious_Threat; End_Status; Other
v1	84	Natural_Threat; Malicious_Threat; End_Status; Event; GPE; DATE; Affected; CARDINAL; PERCENT
v2	84	All entities except: Attacker; Accidental_Threat; Duration; Effect; Possible_Solutions
NER_simplified	120 manually modified	Affected; CARDINAL; PERCENT

7.1.2 RE test cases

Once the findings on the impact of entities and data scarcity on NER were obtained, it was possible to study some RE test cases to determine if this task was feasible, too. To achieve such goal, different test cases were built: three (called "REmulti" cases) would have taken into account the four most present relations in the dataset, while another would have taken into account only the most frequent. Additionally, two extra test cases were built from the latter by training the model on a simplified dataset, to show how beneficial it is to have a higher samples-text ratio compared to the original 120 news articles (these last three are called "REsingle" cases).

The discussion will start from the RE cases that were trained on the original dataset, which contained the following samples for the four most frequent relations:

- *number of*: 248,
- *ends in*: 225,
- *in*: 159,
- *at*: 97.

It is important to keep in mind that, for simplicity, only pairs of *Natural Threat-End Status* and *Malicious Threat-End Status* were taken into account for the *ends in* relations, since the dataset had to be manually labelled once again. For the description of the relations, it is possible to consult Table 4.3. Moreover, relationships were considered valid only if enclosed in one sentence or, at most, spanning between two consecutive sentences. As it is possible to notice, even the most present relations have a number of examples which is insufficient to achieve good performances. Moreover, since the amount was even lower for the others, it seemed not convenient to build test cases by changing the included relationships, differently from the approach adopted with the entities for the NER test cases.

The analysis of the RE cases was also based on finding the effect of a hyperparameter, called "max distance", which defines the maximum number of words that can separate two entities (referred to as the "head" and the "child") linked by a relation. At first, it was left to its default value $max\_distance = 100$, thus obtaining the *REmulti_100* case. Later, the parameter was decreased to $max\_distance = 50$ (*REmulti_50*) and $max\_distance = 25$ (*REmulti_25* and *REsingle_25*).

Before training the dataset was split as follows:

- Training set: 80%,
- Evaluation set: 10%,
- Test set: 10%,

Finally, the impact of the composition of the dataset was taken under exam. Extracting only the sentences containing the most frequent relation, *number of*, and considering only those where both the head and the child were present, a smaller and simplified dataset was obtained. It is composed only of 171 examples (corresponding also the total number of sentences) which were split into training, evaluation and test sets as for the other test cases. Considering that the sentences were short and the results that had been obtained in the meantime for the previously discussed RE test cases, the "max distance" was firstly set to $max\_distance = 15$ and then it was decided to look for relations only for entities that were closer than 10 words apart, thus for $max\_distance = 9$.

7.2 Results

To better understand the performance obtained from the different experiments, results will be divided by task, i.e. NER and RE, following the structure of Section 7.1. The comparison among experiments (obviously for the same task), will be based on the F1 score (Equation 7.1), considered the best performance evaluator as it is the harmonic mean of precision (Equation 7.2) and recall (Equation 7.3). However, also the latter two scores will be included to have a better picture of the behaviour of the model under different conditions.

$$F\text{-}score = 2\frac{precision \cdot recall}{precision + recall} = \frac{TP}{TP + 0.5(FP + FN)} \tag{7.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{7.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{7.3}$$

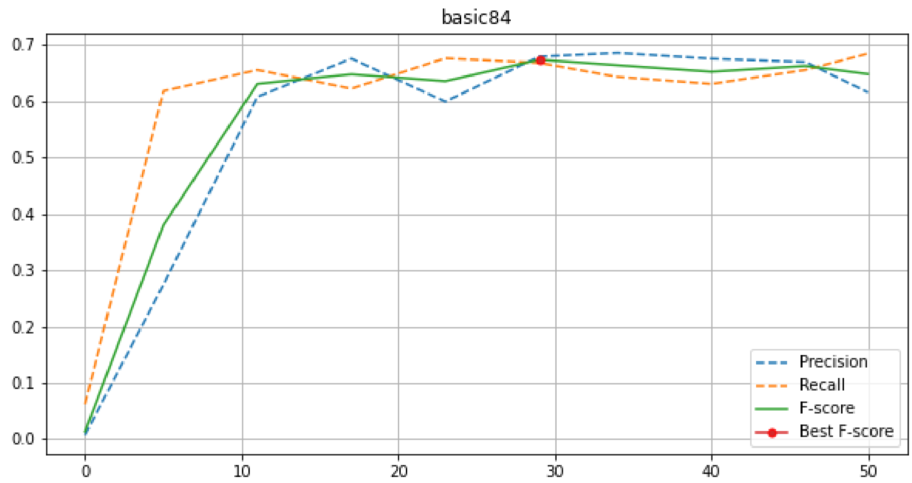
where TP stands for true positive results,
FP for false positive results, and
FN for false negative results.

7.2.1 NER scores

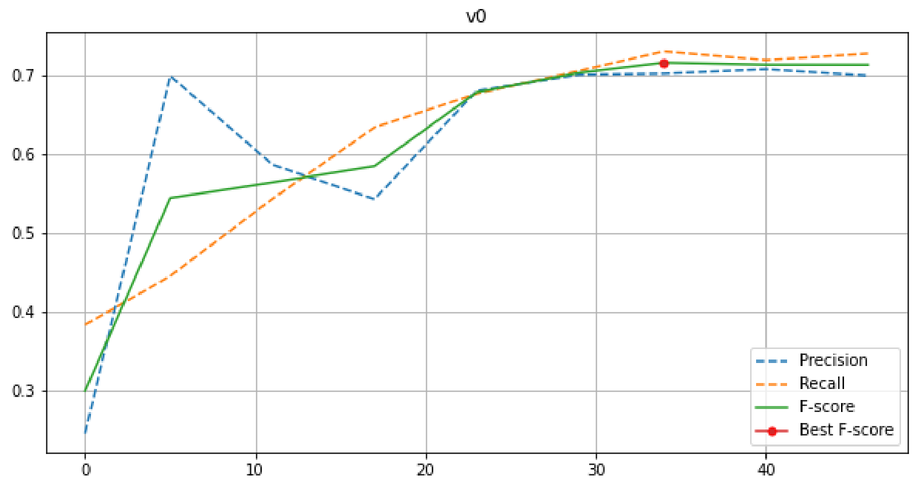
As far as the NER test cases are concerned, it would be better to start the discussion from the "simpler" ones. Considering the experiments run after training the model on the original 84 articles, *basic84* and *v0* show the worst performance (Fig. 7.1). This might seem surprising since it would be logical to expect better scores when the model tries to find a smaller variety of entities. However, it is important to keep in mind that Table 7.2 shows the average scores among all the entities included in the training. Although *v1* and *v2* seem to perform better, when analyzing the result for each entity, it is possible to notice how the precision and recall, and subsequently the F-score, for the classes introduced with the proposed ontology do not show improvement (the most evident example is for the *Natural Threat*, shown in Table 7.3). The reason for such findings is, once again, data scarcity. The increase in performance for the pre-trained experiments is, on the other hand, exactly due to due to those generic purpose entities the spaCy model was already trained for. Scoring higher, they raise the average.

At this point, however, it would be a mistake to consider the experiments a failure. Considering the small number of examples for each entity, the performances are actually in line with the expectations. Moreover, the objective of these tests should never be forgotten or the wrong conclusions would be drawn. The initial question was how increasing the considered portion of the ontology would affect the model. The results prove how including the (almost) complete set of entities is feasible. If the ontology was incorrect (e.g. being confusing, for instance by including different classes covering objects that are too close), a great detriment of performance for the experiments that include more entities would be visible. Further proof of the responsibility of data scarcity for the low scores is that even for the pre-trained entities the performance is not as brilliant as for the model before the new training [45]. Indeed, even if the model was already trained to detect them, the new training does not include enough examples of them to keep the performance at the original level. This is probably why, during the first epochs of training, pre-trained cases *v0*, *v1*, and *v2* show a "spike" in precision, which could be due to the already known entities scoring extremely high thanks to the previous training, followed by a decrease in performance due to this "loss of training" for them but finally stabilizing with the increase in scores for the new ones (Fig. 7.2).

The previous conclusions are confirmed by the behaviour of the model when the dataset is increased. The experiments *basic120* and *misc0* obtained scores comparable to *v1* and *v2*, even if they did not benefit from the inclusion of pre-trained entities (Fig. 7.3). Their higher performance compared to *basic84* and *v0*, which had the same target entities, is due to the larger number of samples.



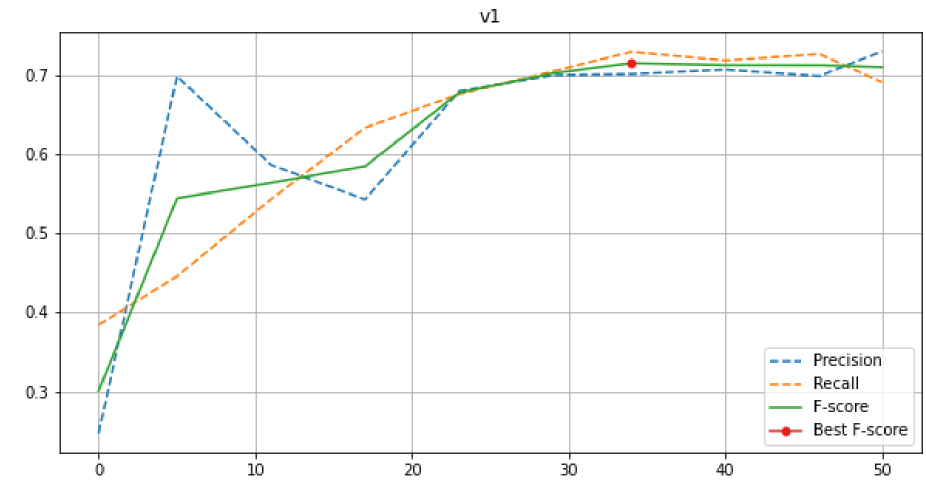
(a) Scores for *basic84*.



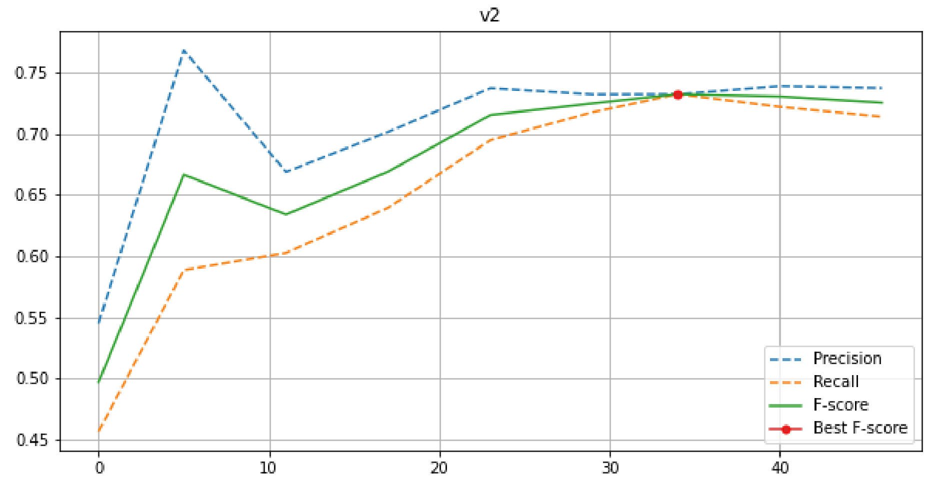
(b) Scores for *v0*.

Figure 7.1: Scores for experiments *basic120* and *misc0*.

Finally, *misc0* presents a higher F-score than *basic120* caused by the addition of the "miscellaneous" entity. This means the inclusion of all the entities that were not present in the samples for *basic120* raises slightly further the average, confirming the findings of Jibbari et al. [13]. For the latter two experiments, a blank model was chosen as base since, predictably, when dealing with custom-designed entities, starting from a pre-trained model does not show any concrete benefit, as noticeable from *v0* not out-performing *basic84*.

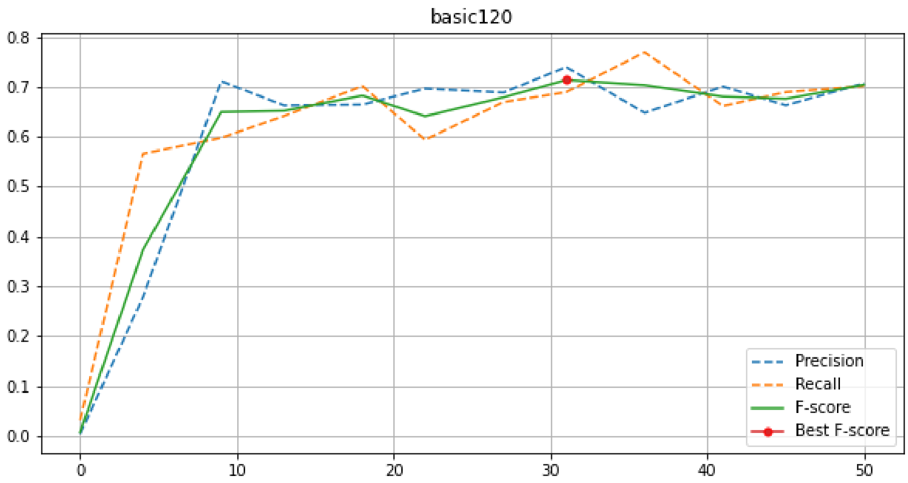


(a) Scores for *v1*.

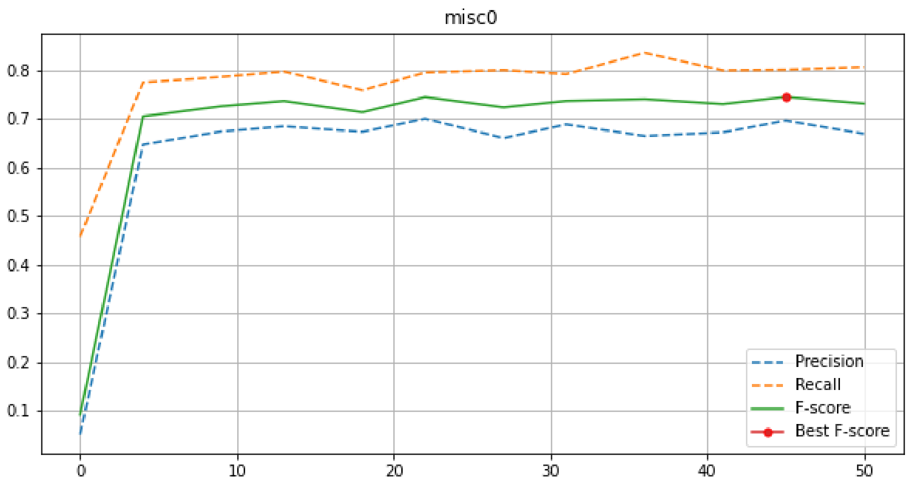


(b) Scores for *v2*.

Figure 7.2: Scores for experiments *basic120* and *misc0*.



(a) Scores for *basic120*.



(b) Scores for *misc0*.

Figure 7.3: Scores for experiments *basic120* and *misc0*.

Finally, regarding *NER_simplified*, the combination of working with two entities (out of three) the model was pre-trained for and the increase of sample-text ratio is the reason behind its good performance.

Table 7.2: Scores for the different NER test cases.

Experiment	F-score	Precision	Recall
basic84	0.67	0.68	0.67
v0	0.68	0.64	0.72
v1	0.72	0.71	0.73
v2	0.73	0.75	0.72
basic120	0.71	0.74	0.69
misc0	0.75	0.69	0.81
NER_simplified	0.86	0.84	0.88

Table 7.3: Scores for *Natural Threat*, comparing the basic test cases and those including entities the NLP model was pre-trained for.

Experiment	Pre-training	*Natural_Threat*		
		F-score	**Precision**	**Recall**
basic84	No	0.60	0.66	0.54
basic120	No	0.64	0.70	0.58
v1	Yes	0.57	0.61	0.53
v2	Yes	0.56	0.64	0.50

7.2.2 RE scores

Considering the experiments for the RE task, the impact of the distance between words linked by a relation was studied. As expected, when the model has to search for relations in a shorter range from an entity, the increase in performance is absolutely not negligible. On average, the scores for *REmulti_25* are 15-20% higher than those of *REmulti_100*, as shown in Table 7.4. Moreover, when the range to check is shorter, the training appears more stable (Fig. 7.4).

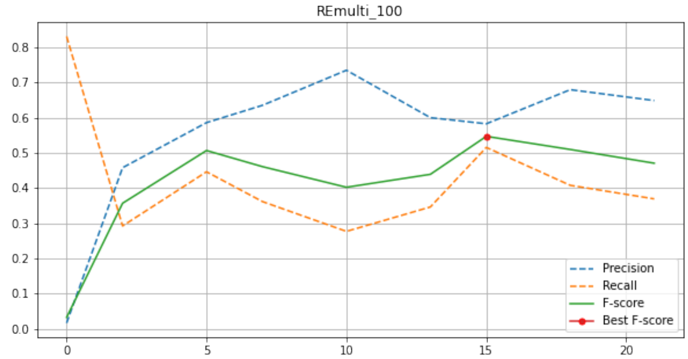
The fact that data scarcity is to be blamed for the not optimal scores is proved also by the results obtained for the experiment with a single relation as objective, *REsingle_25*, trained on the original dataset. If the detriment was due to the ontology, a large performance improvement would be expected when considering only one relation. However, the F-score for *REsingle_25* is not much higher than that for *REmulti_25* (Table 7.4), which was considered for comparison since the two experiments have the same value for *max_distance*.

On the contrary, the improvement is present for *REsingle_15* and *REsingle_9*. After a new version of the dataset was built with a higher samples-text ratio, results showed a significant improvement, so much that *REsingle_9* could almost be considered ready for practical use in on-the-field projects (Fig. 7.6). Moreover, the simplified dataset permitted to train faster, even running a larger number of epochs (Fig. 7.5). However, this came at the cost of manually modifying the dataset and losing the advantages of obtaining the documents automatically from a web scraper. Therefore, it is not a surprise how such operation could not be performed for multiple relations due to time constraints and human resources. Unfortunately, the creation of a corpus is one of the main reasons that might deter the exploitation of NLP in some study areas, such as power systems. Nonetheless, the promising results, especially for the single relations experiments, should be an incentive to invest in this new technology or, at least, conduct further experiments.

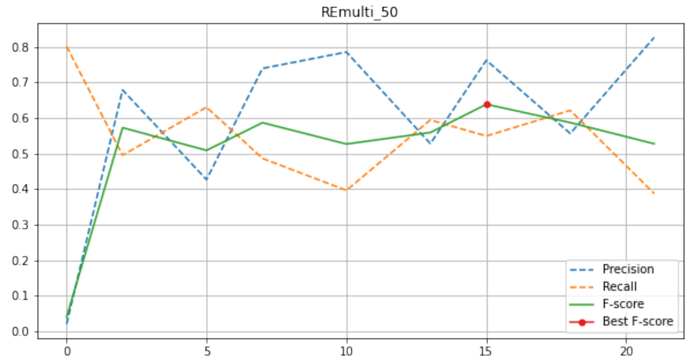
Table 7.4: Scores for the different RE test cases.

Experiment	**dataset**	**“max_distance”**	**F-score**	**Precision**	**Recall**
REmulti_100	original	100	0.55	0.58	0.52
REmulti_50	original	50	0.64	0.76	0.55
REmulti_25	original	25	0.71	0.72	0.71
REsingle_25	original	25	0.75	0.71	0.79
REsingle_15	modified	15	0.85	0.74	1.0
REsingle_9	modified	9	0.91	0.83	1.0

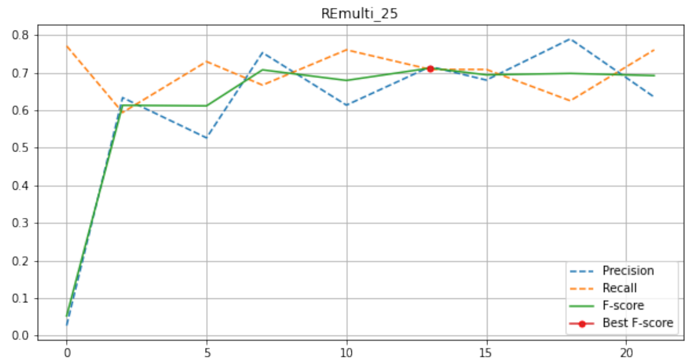
Although these results concern a different task, it is also possible to notice how they are in line with those for NER. Approaching this work with the objective of building an NLP model ready for production would have been irrational. The lack of data, both in quantity and quality (it was not possible to use technical reports, as explained in Section 6.1) obviously presented an obstacle that could not be ignored when setting the goals. Once again, this means that the performance obtained for the extraction of relations should be considered, exactly as for the entity detection, a starting point that could be improved with future works.



(a) Scores for *RE_multi100*.

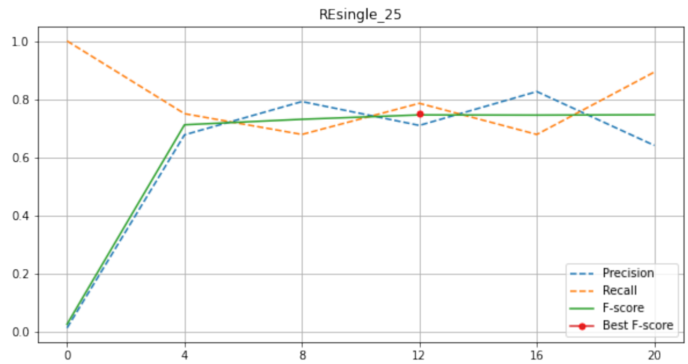


(b) Scores for *RE_multi50*.

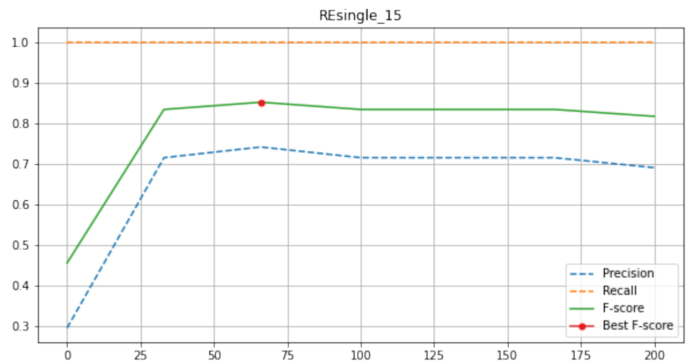


(c) Scores for *RE_multi25*.

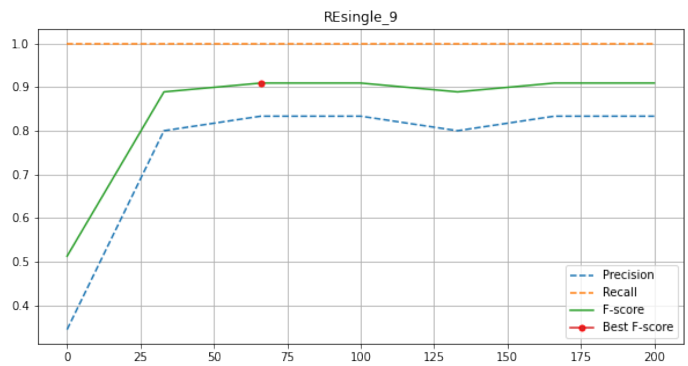
Figure 7.4: Scores for "REmulti" experiments.



(a) Scores for *REsingle_25*.



(b) Scores for *REsingle_15*.



(c) Scores for *REsingle_9*.

Figure 7.5: Scores for the “REsingle” experiments.

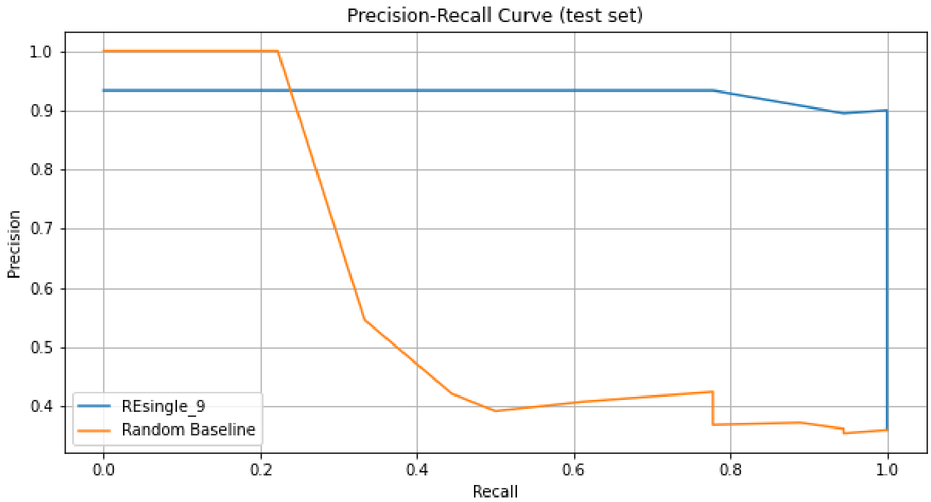


Figure 7.6: Comparison between *REsingle_9* and a random baseline on the test set.

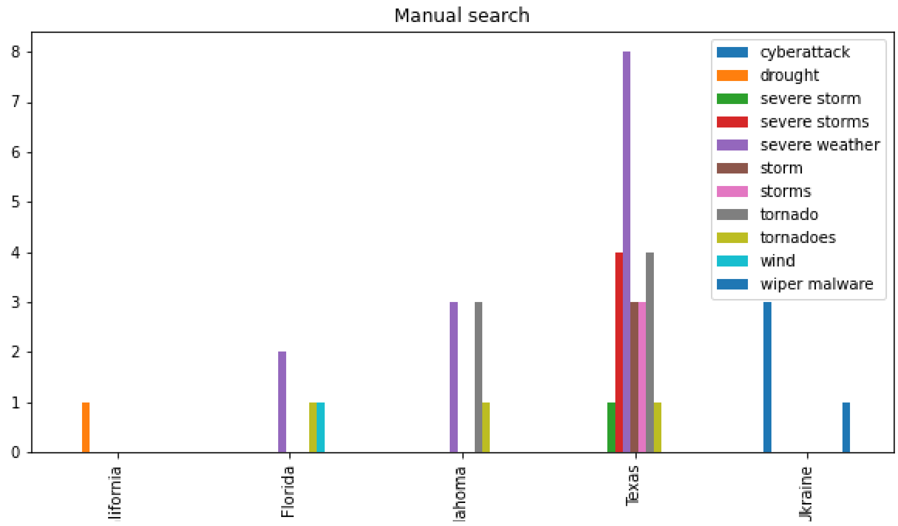
7.3 Possible application

As said in Chapter 2, once the ontology is built, the foundation is laid for the exploitation of NLP in blackout analysis. The applications could be various, from the discovery of the chain of events that lead to blackouts to the creation of databases to automatically keep track of all the power outages in a specific geographic area. For the latter application, a small-scale example will be provided in this section.

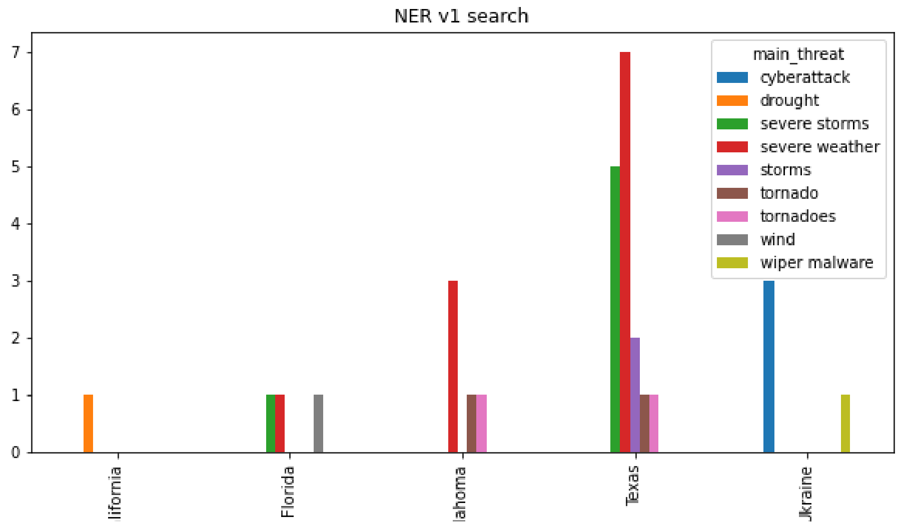
Considering 40 news articles on blackouts in the United States due to natural threats (36) and in Ukraine due to a cyberattack (4), their titles were taken as input for the NER model trained in the *v1* test case. Meanwhile, a human annotator labelled the main threat (the first in order of appearance) and the name of the country/state present in each sample. In the end, the entities detected by the NLP model were compared to those found manually.

It is immediate to notice that the NER *v1* detects entities for almost all the input sentences (Fig. 7.7), despite its not excellent scores obtained after training 7.2, while also considering a good range of different threats.

This proves that NLP tools could truly be beneficial to automatically record blackouts and store data for future studies, especially if optimizing a model for this specific case of use[3].



(a) Results for the manual labelling.



(b) Results for the *v1* search.

Figure 7.7: Geopolitical Entities (GPE) and main threats found in 40 sentences (manual search vs. NER *v1*).

[3] While *v1* was simply trained to show the ontology can be used in practice.

Chapter 8

Conclusions

The project aimed at designing and testing a new ontology specific to blackout analysis by exploiting two NLP tasks, NER and RE. Although the challenges of creating a corpus for this field of use could not be completely overcome, the promising results show that the exploitation of NLP for the extraction of information from human-written documents could prove to be an additional technology to help experts in monitoring smart grids and understanding the complex phenomena behind power outages. This could lead to new solutions and prevention, thus guaranteeing better customer service, while reducing the waste of resources and economic losses.

The experiments that were run proved the solidity of the proposed ontology. The possibility to consider only some of its parts, add new entities and relations or modify the existing ones makes it flexible to the necessities required by new projects and to the challenges that will arise in the future regarding electric energy systems. The test cases that have been discussed and obtained with simple NLP models should be considered as a first step for future studies to optimize NLP tools for more specific cases of use, for instance the automatic creation of databases for blackouts or the discovery of connections between particular events and the electrical grid response. Moreover, the obtained performances should be an incentive to invest in increasing the corpus, possibly through teamwork and interest from grid operators. Alongside customers, they would greatly benefit from the possibility to exploit the technical reports they already have available to support the work of their engineers.

In conclusion, the proposed ontology paves the path to new ways of approaching blackout analysis and its future exploitation is to be hoped, so that this technology could be helpful in ensuring energy access to people even in adverse conditions.

Appendix A

Custom Python script to generate RE train, dev and test sets

content/create_spacy_sets.py

```
# -*- coding: utf-8 -*-
"""
    Create train, dev and test sets for RE (JSON format).
"""

import pandas as pd
import json
import os
import spacy
from math import ceil, floor

TXTPATH = r'txtfiles'
JSONPATH = r'jsonfiles'
SAVEPATH = r'processedfiles'
DBPATH = r'datasets'


def rels_ents(json_path, txt_path):
    '''Load entities from json file.'''

    with open(txt_path, 'r', encoding='UTF-8') as f:
        txt = f.read()


    with open(json_path, 'r', encoding='UTF-8') as f:
```

```python
        data = json.load(f)

    # Divide text into tokens with spaCy.
    nlp = spacy.blank('en')
    all_tokens = nlp(txt)

    # Extract realtions.
    rels0 = tuple(data['_views']['_InitialView']['Relation']) if 'Relation' in data['_views']['_InitialView'] else ()

    # Exctract entities.
    ffs = data['_referenced_fss']
    ents0 = {k: v for k,v in ffs.items() if ffs[k]['_type']=='NamedEntity'}

    # Check if the current document contains relations.
    if not rels0:
        print(json_path, 'has no relations!')
        return None, None, None

    # Save entities in the required structure.
    ents = []
    token_starts = {}
    current = 0
    for k, e in ents0.items():
        start = e['begin'] if "begin" in e else 0
        end = e['end']
        entityLabel = e['value']
        text = txt[start:end]

        current_doc = nlp(text)
        current_substring = nlp(txt[:end])

        current_tokens = len(current_doc)
        tokens_in_substring = len(current_substring)

        token_start = tokens_in_substring - current_tokens
        token_end = token_start + current_tokens

        token_starts[k] = token_start
        ents.append({'text': text, 'start': start, 'end': end, 'token_start': token_start, 'token_end': token_end, 'entityLabel': entityLabel})

    # Save relations in the required structure.
    rels = []
    for r in rels0:

        child_key = str(r['Dependent'])
```

```
        head_key = str(r['Governor'])

        child = int(token_starts[child_key])
        head = int(token_starts[head_key])
        try:
            relationLabel = r['number_of']
        except:
            print(json_path, r)
        rels.append({'head': head, 'child': child, 'relationLabel':
 relationLabel})

    return ents, rels, txt


def create_json(txt, ents, rels, savepath):
    '''Create json file for one news article.'''

    data = {'document': txt, 'tokens': ents, 'relations': rels}
    with open(savepath, 'w', encoding='UTF-8') as f:
        json.dump(data, f)
    return


def create_REset(filespath, setspath):
    '''Join json files of multiple news articles to create train/test
     dataset.'''

    files = os.listdir(filespath)
    data = []

    for file in files:
        with open(filespath+'/'+file, 'r', encoding='UTF-8') as f:
            data.append(json.load(f))

    with open(setspath+r'\dataset.json', 'w', encoding='UTF-8') as f:
        json.dump(data, f)

    return


def train_test_split(setspath):
    '''Split data into train, dev and test sets.'''

    with open(setspath+r'\dataset.json', 'r', encoding='UTF-8') as f:
        data = json.load(f)

    print(len(data))
    train_size = ceil(len(data)*0.8)
    test_size = floor(len(data)*0.1)
```

```python
    print(train_size, len(data)-train_size-test_size, test_size)

    with open(DBPATH+r'\train.json', 'w', encoding='UTF-8') as f:
        json.dump(data[:train_size], f)

    with open(DBPATH+r'\dev.json', 'w', encoding='UTF-8') as f:
        json.dump(data[train_size:train_size+test_size], f)

    with open(DBPATH+r'\test.json', 'w', encoding='UTF-8') as f:
        json.dump(data[train_size+test_size:], f)



if __name__ == '__main__':
    '''Create RE sets.'''

    # Load text files and JSON files (from the UIMA CAS JSON format).
    txtfiles = os.listdir(TXTPATH)
    jsonfiles = os.listdir(JSONPATH)

    txtfiles.sort()
    jsonfiles.sort()

    print(txtfiles[:5])
    print(jsonfiles[:5])

    articles = tuple(zip(jsonfiles, txtfiles))
    print(articles[:5])

    i = 0
    for file in articles:
        filepath = JSONPATH+'/'+file[0]
        savepath = SAVEPATH+'/'+file[0]
        textpath = TXTPATH+'/'+file[1]

        with open(filepath, 'r', encoding='utf-8') as f:

            entities, relations, text = rels_ents(filepath, textpath)

            if relations:
                i+=1

            if not relations:
                continue

            create_json(text, entities, relations, savepath)

    print(i, 'files have the relation(s)')
    create_REset(SAVEPATH, DBPATH)
```

```
    train_test_split(DBPATH)
```

Bibliography

[1] Ettore Bompard, Tao Huang, Yingjun Wu, and Mihai Cremenescu. «Classification and trend analysis of threats origins to the security of power systems». In: *International Journal of Electrical Power & Energy Systems* 50 (2013), pp. 50–64 (cit. on pp. 1, 4, 23).

[2] Paul Hines, Jay Apt, and Sarosh Talukdar. «Trends in the history of large blackouts in the United States». In: *2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century*. IEEE. 2008, pp. 1–8 (cit. on p. 1).

[3] Hassan Haes Alhelou, Mohamad Esmail Hamedani-Golshan, Takawira Cuthbert Njenda, and Pierluigi Siano. «A survey on power system blackout and cascading events: Research motivations and challenges». In: *Energies* 12.4 (2019), p. 682 (cit. on p. 1).

[4] PowerOutage.US. *Major Power Outage Events*. URL: `https://poweroutage.us/about/majorevents` (visited on 02/25/2022) (cit. on p. 1).

[5] City of Austin & Travis County. «2021 Winter Storm Uri After-Action Review: Findings Report». In: (Nov. 4, 2021) (cit. on p. 1).

[6] Salman Masood. «Much of Pakistan Loses Power in Massive Blackout». In: *The New York Times* (Jan. 9, 2021) (cit. on p. 1).

[7] Claudia Wyrwoll. «User-Generated Content». In: *Social Media*. Springer, 2014, pp. 11–45 (cit. on p. 3).

[8] KR1442 Chowdhary. «Natural language processing». In: *Fundamentals of artificial intelligence* (2020), pp. 603–649 (cit. on p. 3).

[9] Jeffrey Ray, Olayinka Johnny, Marcello Trovati, Stelios Sotiriadis, and Nik Bessis. «The rise of big data science: A survey of techniques, methods and approaches in the field of natural language processing and network theory». In: *Big Data and Cognitive Computing* 2.3 (2018), p. 22 (cit. on p. 3).

[10] Kavi Mahesh, Sergei Nirenburg, et al. «A situated ontology for practical NLP». In: *Proceedings of the IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing*. Vol. 19. 1995, p. 21 (cit. on p. 3).

[11] Marcel Robeer, Garm Lucassen, Jan Martijn EM Van Der Werf, Fabiano Dalpiaz, and Sjaak Brinkkemper. «Automated extraction of conceptual models from user stories via NLP». In: *2016 IEEE 24th international requirements engineering conference (RE)*. IEEE. 2016, pp. 196–205 (cit. on p. 3).

[12] Kanix Wang et al. «NERO: a biomedical named-entity (recognition) ontology with a large, annotated corpus reveals meaningful associations through text embedding». In: *NPJ systems biology and applications* 7.1 (2021), pp. 1–8 (cit. on p. 4).

[13] Ali Jabbari, Olivier Sauvage, Hamada Zeine, and Hamza Chergui. «A French corpus and annotation schema for named entity recognition and relation extraction of financial news». In: *Proceedings of the 12th Language Resources and Evaluation Conference*. 2020, pp. 2293–2299 (cit. on pp. 4, 24, 32, 37).

[14] Johan A.K. Suykens and Joos Vandewalle. «Least Squares Support Vector Machine Classifiers». In: *Neural Processing Letters* 9 (1999), pp. 293–300. DOI: 10.1023/A:1018628609742. URL: https://doi.org/10.1023/A:1018628609742 (cit. on p. 4).

[15] Ziling Fan, Luca Soldaini, Arman Cohan, and Nazli Goharian. «Relation Extraction for Protein-Protein Interactions Affected by Mutations». In: New York, NY, USA: Association for Computing Machinery, 2018. ISBN: 9781450357944. DOI: 10.1145/3233547.3233617. URL: https://doi.org/10.1145/3233547.3233617 (cit. on p. 4).

[16] Jake Lever and Steven Jones. «Painless Relation Extraction with Kindred». In: *BioNLP 2017* (2017), pp. 176–183 (cit. on p. 4).

[17] Guilherme Bauer Negrini. *Biomedical text natural language processing (BioNLP) using scispaCy*. 2021. URL: https://gbnegrini.com/post/biomedical-text-nlp-scispacy-named-entity-recognition-medical-records/ (visited on 06/09/2022) (cit. on p. 5).

[18] Xavier Schmitt, Sylvain Kubler, Jérémy Robert, Mike Papadakis, and Yves LeTraon. «A replicable comparison study of NER software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate». In: *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE. 2019, pp. 338–343 (cit. on p. 5).

[19] Matthew Honnibal and Ines Montani. «spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing». In: *To appear* 7.1 (2017), pp. 411–420 (cit. on p. 5).

[20] Matthew Honnibal and Ines Montani. *Training Pipelines & Models*. URL: https://spacy.io/usage/training (visited on 02/21/2022) (cit. on p. 5).

[21] Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. «OpenNRE: An Open and Extensible Toolkit for Neural Relation Extraction». In: *Proceedings of EMNLP-IJCNLP: System Demonstrations*. 2019, pp. 169–174. DOI: `10.18653/v1/D19-3029`. URL: `https://www.aclweb.org/anthology/D19-3029` (cit. on p. 5).

[22] Explosion. *Prodigy*. URL: `https://prodi.gy/` (visited on 06/29/2022) (cit. on p. 6).

[23] Explosion. *Explosion*. URL: `https://explosion.ai/` (visited on 06/29/2022) (cit. on p. 6).

[24] *doccano*. URL: `https://doccano.herokuapp.com/` (visited on 06/29/2022) (cit. on p. 6).

[25] Pontus Stenetorp, Sampo Pyysalo, and Goran Topić. *brat*. URL: `https://brat.nlplab.org/` (visited on 06/29/2022) (cit. on p. 6).

[26] Jan-Christoph Klie. «INCEpTION: Interactive machine-assisted annotation.» In: *DESIRES*. 2018, p. 105 (cit. on p. 6).

[27] IEEE Italy Section and IEEE Region 8. *IEEE MELECON 2022*. URL: `https://melecon2022.org/wp-content/uploads/2022/06/IEEE_MELECON2022_FinalProgram_v2.pdf` (cit. on p. 9).

[28] Jixuan Zheng, David Wenzhong Gao, and Li Lin. «Smart Meters in Smart Grid: An Overview». In: *2013 IEEE Green Technologies Conference (GreenTech)*. 2013, pp. 57–64. DOI: `10.1109/GreenTech.2013.17` (cit. on p. 9).

[29] Juan Shishido and EnerNOC Utility Solutions. «Smart meter data quality insights». In: *ACEEE Summer Study on Energy Efficiency in Buildings*. Vol. 12. 2012, pp. 277–288 (cit. on p. 9).

[30] Robin Lloyd and CNN Interactive Senior Writer. «Metric mishap caused loss of NASA orbiter». In: *CNN Interactive* (1999) (cit. on p. 11).

[31] Nicola Guarino, Daniel Oberle, and Steffen Staab. «What is an ontology?» In: *Handbook on ontologies*. Springer, 2009, pp. 1–17 (cit. on p. 11).

[32] Tom Gruber. *Ontology*. 2018 (cit. on p. 11).

[33] Matthew Honnibal and Ines Montani. *Chapter 4: Training a neural network model*. URL: `https://course.spacy.io/en/chapter4` (visited on 02/21/2022) (cit. on pp. 11, 24, 27).

[34] Janez Brank, Marko Grobelnik, and Dunja Mladenic. «A survey of ontology evaluation techniques». In: *Proceedings of the conference on data mining and data warehouses (SiKDD 2005)*. Citeseer Ljubljana Slovenia. 2005, pp. 166–170 (cit. on p. 13).

[35] Mark A Musen. «The protégé project: a look back and a look forward». In: *AI matters* 1.4 (2015), pp. 4–12 (cit. on p. 16).

[36] Denny Britz. *Attention and Memory in Deep Learning and NLP*. Dec. 27, 2016. URL: https://opendatascience.com/attention-and-memory-in-deep-learning-and-nlp/ (visited on 06/09/2022) (cit. on p. 17).

[37] Christopher Olah. *Understanding LSTM Networks*. Aug. 27, 2015. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (visited on 06/09/2022) (cit. on p. 17).

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. «Attention is All you Need». In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf (cit. on pp. 17–19).

[39] Ajay Halthor. *CodeEmporium*. URL: https://www.youtube.com/c/CodeEmporium/ (visited on 11/04/2021) (cit. on p. 21).

[40] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. «Bert: Pre-training of deep bidirectional transformers for language understanding». In: *arXiv preprint arXiv:1810.04805* (2018) (cit. on p. 22).

[41] Explosion. *Model Architectures*. URL: https://spacy.io/api/architectures (visited on 11/04/2021) (cit. on p. 22).

[42] *PDF.CO*. URL: https://pdf.co/ (visited on 10/04/2021) (cit. on p. 23).

[43] Walid Amamou. *Relation Extraction Transformer*. July 27, 2021. URL: https://github.com/walidamamou/relation_extraction_transformer (visited on 03/19/2022) (cit. on p. 24).

[44] Bharat Govind Gautam and Bernadette Baum. «La Niña to emerge in Aug-Oct season, U.S. weather service says». In: *Reuters* (Aug. 12, 2021). URL: https://www.reuters.com/business/environment/us-forecaster-sees-la-nia-emerging-autumn-2021-2021-08-12/ (cit. on p. 25).

[45] Matthew Honnibal and Ines Montani. *English - spaCy Models Documentation*. URL: https://spacy.io/models/en (visited on 02/21/2022) (cit. on p. 36).