



HAL
open science

Text mining for automating TRIZ-based inventive design process using patent documents

Guillaume Guarino

► To cite this version:

Guillaume Guarino. Text mining for automating TRIZ-based inventive design process using patent documents. Computer science. Université de Strasbourg, 2022. English. NNT : 2022STRAD033 . tel-04047047

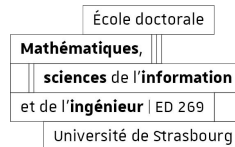
HAL Id: tel-04047047

<https://theses.hal.science/tel-04047047>

Submitted on 27 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNIVERSITÉ DE STRASBOURG
ED269
ICube

DOCTORAL THESIS

Text mining for automating TRIZ-based Inventive Design process using patent documents

Author:
Guillaume GUARINO

A thesis submitted for the obtention of the degree of
Doctor of Philosophy
in
Computer Science

Thesis publicly defended on 09/12/2022 before the jury composed of:

Thesis Director:	Denis Cavallucci	Professor, INSA Strasbourg
Supervisor:	Ahmed Samet	Associate Professor, INSA Strasbourg
Referee:	Gül E. Kremer	Professor, University of Dayton
Referee:	Jean-Charles Lamirel	Associate Professor (HDR), Université de Strasbourg
Examiner:	Sihem Amer-Yahia	CNRS Research Director, Université Grenoble Alpes
Examiner:	Pierre Gañçarski	Professor, Université de Strasbourg
Examiner:	Alexandre Termier	Professor, Université de Rennes 1

UNIVERSITÉ DE STRASBOURG

Abstract

Université de Strasbourg
Institut National des Sciences Appliquées Strasbourg (INSA Strasbourg)

Doctor of Philosophy

Text mining for automating TRIZ-based Inventive Design process using patent documents

by Guillaume GUARINO

The innovation level of a product is one of the key indicators that influences the way this product is positioned among its competitors. This level of innovation can be measured through indicators including the performance of the product in its primary function, the environmental impact or the cost. Increasing the innovation level is one of the toughest challenges. The Theory of Inventive Problem Solving (TRIZ) was proposed to push for disruptive innovation and maximize the chances of solving a problem in an inventive manner. Altshuller, the founder of this theory, presented a pseudo-algorithm going from the formulation of a problem to its resolution in order to help engineers. His theory is based on a very strong observation: each problem in a domain A can be linked to a problem already solved in a domain B, whose solution would only have to be adapted.

However, finding a corresponding problem in another domain requires, either reading all the scientific publications and patents that exist, or developing an algorithm capable of characterizing the problems solved by each existing solution. In this thesis, we investigated this second research direction.

In order to enable a cross-domain search for solutions, neural-based approaches are developed to identify the key concepts of TRIZ theory in patents. In particular, we propose a contradiction mining approach based on automatic summarization, a parameter mining approach integrating the syntactic structure of sentences and a unified multi-task model showing unequalled performances. Real-world datasets corresponding to the different tasks are also introduced. Our best approach is able to mine 46% of the contradictions in our dataset. Software demonstrators are built to show the viability and efficiency of our approaches.

Keywords: Natural Language Processing (NLP), Patent Mining, Neural Network, Conditional Random Field, Theory of Inventive Problem Solving (TRIZ)

Acknowledgements

First of all, I would like to thank my thesis director Denis Cavallucci. His passion for TRIZ is really inspiring. I was not expecting to study in depth this theory but the more I learn about it the more I am attracted to it. I was able to take advantage of total freedom of action to create my own path in this very large and motivating field of application with its universal scope.

Special thanks to Ahmed Samet who always found the time to discuss my approaches, read, comment and find new ideas to improve my papers. He was always there when I needed help. I also want to thank him for the courses I was able to give throughout my thesis, especially the machine learning and embedded systems course which allowed me to dig into subjects that I was not always an expert on and to deepen my knowledge.

A big thank you to Cloé Urban, who, during these three years, despite the distance and a total absence of answers, continued to talk to me and support me in my thesis. A thought for Milo whose passion for computer science encouraged me to persevere in my research. He also accompanied me during those long evenings of programming.

I also want to thank my family who always pushed me to do better and whose presence appeases me.

Finally, I would like to thank the doctoral students and colleagues that I have met during these years, always in a good mood. I also want to mention all my faithful interns who helped me create my datasets.

Contents

Abstract	iii
Acknowledgements	v
Introduction	1
General Background	1
Motivation	3
Contributions	3
Structure of the thesis	4
1 Neural Network-based approaches for Text Mining	5
Introduction and overview of data mining processes	5
1.1 Tokenization	6
1.2 Word Embedding models	7
1.2.1 Non-contextual embeddings	7
Word2Vec	8
GloVe	9
fastText	9
Discussion	9
1.2.2 Contextual embeddings	10
Recurrent Neural Networks (RNN)	10
Transformers	11
Convolutional Neural Networks (CNN)	14
1.3 Targeted content identification and extraction	15
1.3.1 Document classification	16
1.3.2 Named Entity Recognition (NER)	16
1.3.3 Summarization through sentence classification	16
1.3.4 Summarization through generation	17
Conclusion	20
2 TRIZ and Inventive Design in patent documents	21
Introduction	21
2.1 TRIZ theory	21
2.2 Patent Mining	24
2.2.1 Patent structure	24
2.2.2 Available datasets	25
2.2.3 Patent search	26
2.2.4 Patent classification	27
2.2.5 Generation tasks	27
2.3 TRIZ and patent mining	27
2.3.1 Contradictions and TRIZ parameters extraction	28
2.3.2 Inventive principles, physical effects and solution mining	29
2.3.3 Estimation of inventiveness level	30

Summary and conclusion	30
3 Extractive summarization for TRIZ contradiction mining	35
Introduction	35
3.1 Dataset	35
3.1.1 Parts of interest	35
3.1.2 Labeling details	36
3.2 Extractive Approach	37
3.2.1 Baseline approach: SummaTRIZ	37
3.2.2 PaGAN: Multi-level patent classification and semi-supervised contradiction mining	38
Document-based approach for contradiction mining	39
Semi-supervised contradiction mining	40
3.3 Experiments	43
3.3.1 Metrics	43
3.3.2 Hardware and experimental details	43
3.3.3 Results	47
3.3.4 Case study	49
Positive examples	49
Negative examples	49
Conclusion	50
4 Syntactic Conditional Random Field for TRIZ Parameter mining	51
Introduction	51
4.1 Conditional Random Fields and TRIZ parameter mining	52
4.2 Matrix-based Syntactic CRF	54
4.2.1 SynCRF-matrix	55
4.2.2 SynCRF-point	55
4.3 Potential-based Syntactic CRF	57
4.3.1 SynCRF-pos: Part Of Speech-based Syntactic CRF	57
4.3.2 SynCRF-context: context-based Syntactic CRF	58
4.4 Dataset and training	59
4.5 Experiments and results	60
4.5.1 Metrics and baseline approaches	60
4.5.2 Results	61
SynCRF-matrix	61
SynCRF-point	64
SynCRF-pos	64
SynCRF-context	64
Comparison with the state of the art	64
4.5.3 Discussion	65
Conclusion	66
5 TRIZ Multi-task model	67
Introduction	67
5.1 Naïve model and SummaTRIZ demonstrator	68
5.1.1 Module-based naïve model	68
5.1.2 Case study with SummaTRIZ demonstrator	68
5.2 Multi-task model	72
5.2.1 Backbone and shared parameters	72

5.2.2	Abstractive training and TRIZ Evaluation Parameters-Aware Attention (TEP2A)	74
	TEP2A	74
	Abstractive training and loss	75
5.2.3	Validation	75
5.2.4	Ideas Demonstrator	76
5.2.5	Interpretation of Inventive Principles	77
5.3	Experiments	77
5.3.1	Comparison between Naïve and Multi-task models	77
5.3.2	Parameters attention and modified loss	83
	Conclusion	83
	Conclusion and perspectives	85
	A work in line with the current era	85
	Contributions	85
	Perspectives	86
	A Demonstrators	89
A.1	Patent scrapper and database	89
A.2	Softwares and computing	90
A.3	Summatriz (Naïve model, domain mapping)	90
A.3.1	Load patents or make a query to retrieve patents	90
A.3.2	Analyze patents	91
A.3.3	Optional: correct the results	91
A.4	Ideas (Multi-task model, problem graph generation)	92
A.4.1	Starting a project	92
A.4.2	Selecting the data	92
A.4.3	Creating the problem graph	92
	B Neural Networks	97
B.1	Formal neuron and neural networks	97
B.2	Training	99
B.2.1	Loss	99
B.2.2	Error back-propagation	100
B.3	Training and specialization	101
	Author publications	103
	Computer Science	103
	Innovation	104
	Bibliography	105

List of Figures

1	A portion of TRIZ matrix (the original is 39*39)	2
2	TRIZ stages for problem solving	2
1.1	Data mining process	6
1.2	Word2Vec (Mikolov et al., 2013b) models	8
1.3	Structure of a recurrent network	10
1.4	Structure d'un réseau Long Short Term Memory	11
1.5	BERTDevlin et al., 2019 specialization for downstream tasks	12
1.6	BERTDevlin et al., 2019's Attention Mechanism	13
1.7	Self attention example	13
1.8	Convolution in a CNN	15
1.9	Summarization network	17
1.10	Decoder with BERT(Devlin et al., 2019) encoder	18
1.11	Unified text-to-text framework (T5)	19
2.1	TRIZ stages for problem solving	24
2.2	Problems and partial solutions represented in a graph	28
2.3	Methodology for problem solving according to (Liang et al., 2009)	29
2.4	Specialisation of TRIZ matrix (Wang et al., 2016)	30
3.1	Baseline approach: SummaTRIZ (Guarino et al., 2020)	38
3.2	PaGAN architecture	41
4.1	Impossible label transitions	52
4.2	Multiple indexed transition matrices for POS information integration	55
4.3	Pointing mechanism to integrate POS information in label prediction	56
4.4	SynCRF-pos architecture for POS-adapted pairwise potentials generation	58
4.5	SynCRF-context architecture	59
5.1	Naïve approach	69
5.2	Process used in the case study to represent LCD screens main contradictions/research topics	70
5.3	TRIZ contradictions occurrences in LCD screens SoTA	70
5.4	Multi-task model	73
5.5	Distribution of inventive principles in the dataset	74
5.6	Example of problem graph within Ideas for a scooter	77
5.7	Example of problem graph built automatically	78
A.1	SummaTRIZ: searching for patents	90
A.2	SummaTRIZ: uploading patents and validation	91
A.3	SummaTRIZ: analyzing data	91
A.4	SummaTRIZ: results	93
A.5	Ideas: creating a project	94
A.6	Ideas: defining a project	94

A.7	Ideas: searching for patents and extraction	95
A.8	Ideas: patents selection	96
A.9	Ideas: extraction and problem graph	96
B.1	Perceptron	98
B.2	Neural Network	99
B.3	Gradient descent	100

List of Tables

2.1	Separation principles	23
2.2	Inventivity Levels	24
2.3	TRIZ parameters	25
2.4	TRIZ inventive principles	26
2.5	CPC sections	26
2.6	Patent Indicators and Explanations	30
2.7	Summary of the state of the art in TRIZ domain	32
2.8	Summary of the possible approaches	33
3.1	Details on the summarization dataset	37
3.2	Sentence classification 1 (with LSTM generator)	44
3.3	Sentence classification 2 (with LSTM generator)	44
3.4	Document classification (with LSTM generator)	45
3.5	Sentence classification 1 (with ANN document classifier)	45
3.6	Sentence classification 2 (with ANN document classifier)	46
3.7	Document classification (with ANN document classifier)	46
4.1	Details on the parameter dataset	60
4.2	Matrix-based SynCRF results with BERT encoding	62
4.3	Potential-based SynCRF results with BERT encoding	62
4.4	Potential-based SynCRF results with XLNet encoding	63
4.5	Comparison of SynCRF with the state of the art	63
5.1	EPs mapped on TRIZ EPs for 18/12 contradiction	71
5.2	EPs mapped on TRIZ EPs for 18/22 contradiction	71
5.3	Identified Inventive Principles in patent US10723403	78
5.4	Multi-Task model vs Naïve model (EP/AP)	79
5.5	Multi-Task model vs Naïve model (Contradictions)	79
5.6	Multi-Task model vs Naïve model (Validation)	79
5.7	Multi-Task model (Solution Concepts / Inventive Principles)	79
5.8	Multi-Task model variants (EP/AP)	82
5.9	Multi-Task variants (Contradictions)	82
5.10	Multi-Task variants (Validation)	82
5.11	Multi-Task variants (Solution Concepts / Inventive Principles)	82

Glossary

action parameter Parameter of a technical system with respect to which the designer has a power of modification of state. This type of parameter generally has two opposite directions that can potentially bring a benefit to the object.

attention mechanism Neural technique meant to mimic cognitive attention. It enhances some part of the input data and diminishes some other part depending on the context.

classifier Machine learning algorithm used to assign a class label to a data input.

Conditional Random Field Graphical model often applied in pattern recognition and machine learning and used for structured prediction. It models dependencies between labels through a transition matrix.

contradiction A domain-free formulation of a problem. A contradiction is composed of two evaluation parameters. The improvement of one of the evaluation parameters through the modification of an action parameter leads to a degradation of the other.

decoder Machine Learning model which generates an interpretable output (sound, image, text, ...). It usually utilizes the features extracted by an encoder.

encoder Machine Learning model which maps data (texts, images, ...) onto a multi-dimensional space.

evaluation parameter Parameter of a technical system whose nature lies in the ability to evaluate the positive and negative aspects resulting from a choice of the designer. This type of parameter has only one logical direction of progress, the other direction seems aberrant.

Generative Adversarial Network Two competing neural networks called generator and discriminator. The generator generates data close to a target distribution. The discriminator takes as input both generated data and real data samples. It must recognize which data are real and which data were generated. The generator must fool the discriminator. They are both trained together.

inventive principle 40 methods proposed by Altshuller to solve a contradiction. Inventive principles are very general formulations of solutions (Segmentation, The other way around,...) . They must be applied to each particular problem and context.

pairwise potentials See **transition matrix**.

Part Of Speech Grammatical category of a word. Part-Of-Speech tagging is a well-known task in Natural Language Processing.

transformer Neural network composed of several stacked attention mechanisms.

transition matrix Matrix of trainable parameters in a Conditional Random Field (see CRF). It contains pairwise potentials which model label dependencies. These potentials refer to the probabilities of giving a certain label to the token T_i considering the label given to the preceding token T_{i-1} .

TRIZ parameters 39 parameters defined by Altshuller in TRIZ theory. These parameters can describe any problem from any domain through a contradiction between two of them.

Introduction

Contents

General Background	1
Motivation	3
Contributions	3
Structure of the thesis	4

General Background

Invention traditionally relies on expert exchanges, brainstorming and ideas shared on a whiteboard. The promise of this work is to automatically populate this board with solution concepts.

Improvements have been made to the process of ideation of new products with, in particular, concurrent engineering or Integrated Product Development (IPD) which consists in studying, in parallel, all the issues related to a new product (design, manufacturing, marketing opportunities) during its development process (Kusar et al., 2004; Prasad, 1996a; Prasad, 1996b; Prasad, Wang, and Deng, 1998). However, even if these techniques allow selecting the best solution concepts or accelerating the innovation process, the quality and the experience of the engineers are still the pillar on which we rely for finding the solution concepts.

The TRIZ theory (acronym for Theory of Inventive Problem solving in Russian) was developed in the post-World War II period by an engineer of the ex-USSR, Genrich Altshuller. His idea was to develop a method enhancing innovation capabilities dedicated to people when solving a problem (Altshuller, 1984; Savransky, 2000). It relies on exploiting problem-solutions couples from two distant technical domains. Altshuller then, among other tools, introduced his TRIZ matrix (Figure 1), which is now well known in the innovation ecosystem.

This matrix builds links between problems and solutions. The study of thousands of patents in all fields led by Altshuller highlighted the important proximity between the inventive paths in all fields. He compiled statistics on the techniques used to invent (which he called inventive principles) for each type of problem (which he formulated as a TRIZ contradiction). An inventive principle is a phrase composed of a few words presenting a general action which could be applied to solve the problem (for instance: segmentation, asymmetry, intermediary etc.). A contradiction is a domain-free formulation of a problem. When comparing problems from different fields, the arising problem is the vocabulary used to describe the problems, which is fundamentally different for a problem in mechanics and a problem in chemistry. Thus, Altshuller introduced general parameters (called TRIZ parameters) which can apply to all domains to describe problems (Weight of Stationary Objects, Speed, Power, Waste of Energy...). A contradiction from TRIZ domain is seen as a parameter improvement which leads to another parameter degradation and compromising between these two parameters is not the appropriate path to invent. One

39 Engineering Parameters		39 Engineering Parameters						40 Inventive Principles
		Weight of moving object	Weight of stationary object	Length of moving object	Length of stationary object	Area of moving object	Area of stationary object	
1	Weight of moving object			15, 8, 29, 34		29, 17, 38, 34		1. Segmentation
2	Weight of stationary object				10, 1, 29, 35		35, 30, 13, 2	2. Extraction
3	Length of moving object	8, 15, 29, 34				15, 17, 4		3. Local Quality
4	Length of stationary object		35, 28, 40, 29				17, 7, 10, 40	4. Asymmetry
5	Area of moving object	2, 17, 29, 4		14, 15, 18, 4				5. Merging
6	Area of stationary object		30, 2, 14, 18		26, 7, 9, 39			6. Universality
	.							.
	.							.
	.							.

FIGURE 1: A portion of TRIZ matrix (the original is 39*39)

should find a solution that both improve the first parameter while also improving the second. The matrix gathers the inventive principles (i.e. the inventive paths to follow) that statistically have the best chance to succeed not to compromise for each possible contradiction between TRIZ parameters. The inventive solving process (Figure 2) follows these steps:

- A problem is identified.
- Generalization: the problem is formulated as a contradiction between TRIZ parameters.
- General Solution: the inventive principles corresponding to the contradiction are drawn from the matrix.
- Specialization: Inventive principles are adapted to the original problem and situation to implement inventive solutions.

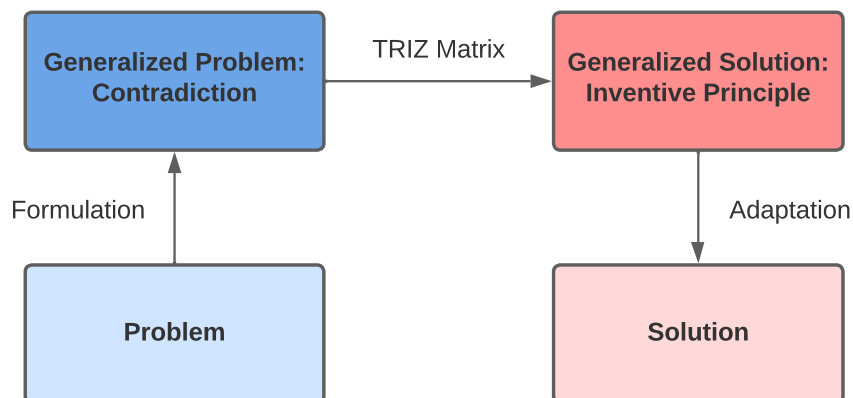


FIGURE 2: TRIZ stages for problem solving

Motivation

Despite this simplified inventive process, inventors still have to interpret the inventive principles proposed by the TRIZ matrix. Thus, the problem of the dependence of the innovation process on experience or even luck is simply shifted to the interpretation of inventive principles. The question then arises of taking a step back and giving concrete examples of applications of inventive principles to facilitate this interpretation. The purpose of this work is, thus, to explain how inventive principles could be used in real life and to directly search for solution concepts that could be applicable. To do so, data mining algorithms are used to characterize the solutions found in patents. This characterization consists in associating to each patent a type of problem (i.e contradiction) which can be solved in order to make a direct link between a problem and possible solutions lying in the patent claims. A database can be filled with patents indexed using the solved contradiction and the way to solve it (inventive principle) to provide a huge and easily exploitable data source for inventions.

Contributions

Our work provides several scientific multidisciplinary contributions at the crossroads of data mining, natural language processing and TRIZ. These contributions are:

- A summary-based approach to mine solved contradictions from patents. Contradictions are first seen as contradictory sentences. The parameters in a first sentence are improved while the parameters in the second sentence are degraded. Contradiction mining therefore relates to a 3-class classification problem. Multi-level classification and semi-supervised learning are explored.
- A sentence-level contradiction dataset was labeled to train and to evaluate the summarization model. The dataset contains 1600 US patents with labeled sentences and 1600 patents without contradictions.
- A token-based classification model for parameters mining. The purpose is to mine the parameters contained in the contradiction sentences. A Conditional Random Field (CRF) on top of a neural encoder is used. A new type of CRF generating pairwise potentials from syntactic patterns is proposed.
- A token-level parameter dataset is labeled to train and to evaluate the parameter models.
- A multi-level model to unify contradiction and parameter mining. The model combines both sub-modules in a full framework starting from patent scraping to results storing.
- A multi-task model for contradiction, parameter and inventive principle mining. This model is trained on all possible TRIZ tasks to optimize the number of parameters and simplify the analysis process. A new TRIZ parameters-aware attention mechanism for contradiction mining is proposed.

Structure of the thesis

This thesis provides 5 chapters including state of the art and contributions. These chapters are listed below:

- **Chapter 1: Neural Network-based approaches for Text Mining** covers the state of the art in text mining and the full process needed to mine large amounts of data. Particular emphasis is given to deep neural networks to build very rich token representations.
- **Chapter 2: TRIZ and Inventive Design in patent documents** discusses TRIZ and Inventive Design Methodology. Work dealing with patent analysis in the context of TRIZ is also highlighted.
- **Chapter 3: Extractive summarization for TRIZ contradiction mining** presents the contributions related to summary and contradiction mining. The labeled dataset is also presented.
- **Chapter 4: Syntactic Conditional Random Field for TRIZ Parameter mining** includes contributions related to parameter mining and CRF enhanced by syntactic information.
- **Chapter 5: Multi-task model for contradictory parameters and solution concept mining** presents unified models for the extraction of contradictions and parameters. A multi-task model integrating inventive principles in addition to contradictions is also introduced.
- **Appendix** presents the software demonstrators and summarizes the history and functioning of a neural network, learning techniques, error back-propagation.

Chapter 1

Neural Network-based approaches for Text Mining

Contents

Introduction and overview of data mining processes	5
1.1 Tokenization	6
1.2 Word Embedding models	7
1.2.1 Non-contextual embeddings	7
1.2.2 Contextual embeddings	10
1.3 Targeted content identification and extraction	15
1.3.1 Document classification	16
1.3.2 Named Entity Recognition (NER)	16
1.3.3 Summarization through sentence classification	16
1.3.4 Summarization through generation	17
Conclusion	20

Introduction and overview of data mining processes

Text mining refers to the process of identifying, enumerating, and analyzing syntactic and semantic characteristics of a text corpus. The purpose of text mining is to bring to light unknown facts, characteristics, patterns and ultimately lead to a better understanding of textual contents. It is one of the main disciplines of artificial intelligence. The data can have a constant structure. In this case we speak of "structured data". If it is not the case, we speak of unstructured data.

The main steps of our process to identify interesting textual contents in technical documents is displayed in Figure 1.1. Several Natural Language Processing (NLP) techniques are employed to mine textual contents, especially classification-based tasks like Named Entity Recognition (NER) and extractive summarization. As neural networks obtain the best results in these tasks, we will focus on neural-based approaches.

A data mining pipeline generally consists of several steps from data pre-processing to a choice of model architecture and results analysis. Textual data must be transformed into numeric data to be processed by a computer. First, the tokenization splits the input document into words. Vector representations of the documents or words, called *embeddings*, can then be built. The embeddings may finally be used to mine the targeted information.

In this chapter, the main approaches for building rich representations of textual content are reviewed. The main tasks allowing precise information extraction or

content characterization in a given framework are presented. Among these, we find automatic summarization, document classification or token classification.

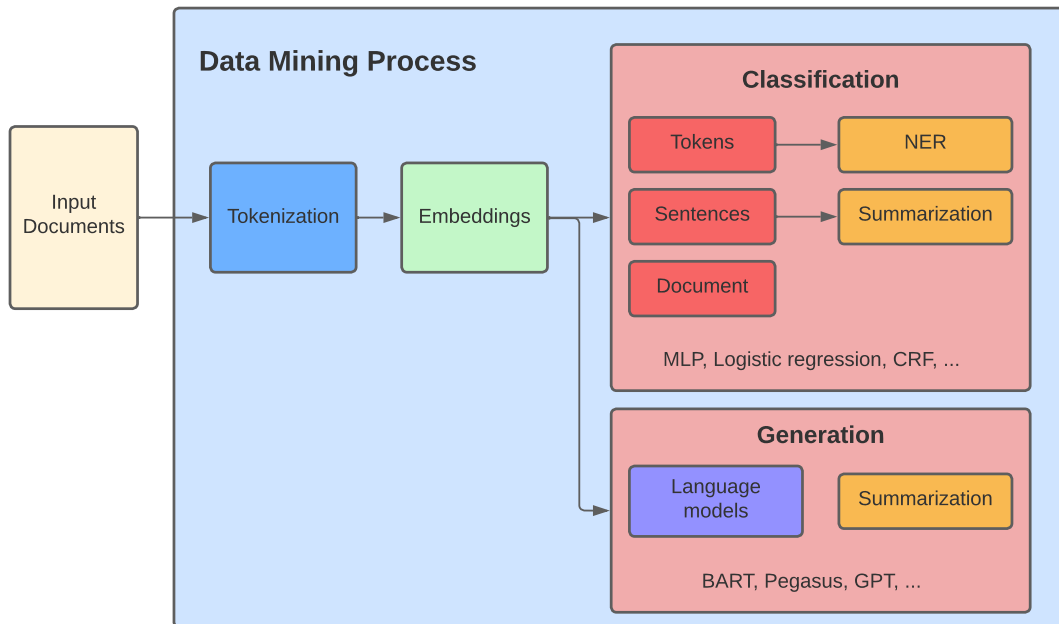


FIGURE 1.1: Data mining process

1.1 Tokenization

The tokenization consists in splitting up a character sequence into pieces called tokens. These tokens correspond to words, sub-words, roots or endings. Splitting the sentence into small elements makes the analysis finer. Tokenization is mandatory in any Natural Language Processing process. Punctuation is often abandoned during tokenization because it is not essential to understand the words' meaning. In this section, we mainly focus on one of the most famous tokenizer designed to handle rare words, which is important for a patent analysis.

A tokenization can be word, character, or sub-word-based. If the tokenization is based on words, it means that a vocabulary with a finite number of words is created (all the words in the original corpus used to "train" the tokenizer if it is trainable). It therefore cannot handle out-of-vocabulary words. Character-based tokenizations can handle all existing words but they do not give sufficient information on the meaning. Sub-word tokenizers do not have these disadvantages. WordPiece (Wu et al., 2016) is currently used with the last state of the art NLP models. It is based on Byte-Pair Encoding (Sennrich, Haddow, and Birch, 2016). Byte-pair encoding (BPE) is a compression algorithm which gathers the most frequent combinations of letters in words to create sub-words. These sub-words constitute what is called the "vocabulary". BPE is an iterative greedy algorithm but once the vocabulary is chosen, its use is straightforward. Byte-pair encoding ensures that the most common words are represented in the vocabulary as a single token. Rarer words (and even unknown words) are broken down into sub-words. A dictionary with ids assigned to each sub-word is created. When tokenizing a textual content the words are cut, using the sub-words in the vocabulary, and replaced with their ids (index of the sub-word in the vocabulary). For example, "The car stops." is mapped to [1, 2, 3] if the

indexes of "The", "car" and "stops" are 1, 2 and 3 in the vocabulary. The output of WordPiece is thus an id sequence which can finally be fed into a neural network or another machine learning algorithm.

1.2 Word Embedding models

After preparing the textual data process through the tokenization, the following step is the analysis of word and sentence meaning. For this, a token, sentence or document is converted into a numeric sequence representing its meaning. These vector representations are called *embeddings*. The underlying principle is that entities with similar meanings have similar representations.

The embeddings construction is linked to the tokenization process. Each token is processed. A document or sentence-based embedding can be built using a *Bag of Words* (a vector representation containing the tokens frequencies) which is the simplest possible embedding. For example, the following sentence "The car hits the wall" can be represented with [2, 1, 1, 1] considering that the number of occurrences of "the" is 2 and 1 for the other words. The vocabulary would be *the, car, hits, wall*. The problem is that these embeddings fail to encapsulate rich semantic meaning and they do not take into account sequentiality in the data. Word or sub-word embeddings are often preferred. Word-based tokenization is often linked to Word2Vec (Mikolov et al., 2013b)-like models (Part 1.2.1). Sub-word-based tokenizers are used by almost all the deep neural networks (Part 1.2.2) which establish the state of the art in the main NLP tasks (Lewis et al., 2020a; Devlin et al., 2019; Yang et al., 2019a). They allow processing unknown words while guaranteeing an optimal integration of information on the structure of the word (root, prefix, suffix, etc.). These neural networks build "contextual" token representations i.e., the token representations are interdependent and different for a same token in different contexts.

In this section, word embedding models are presented. They were developed to build the best representations of the words in terms of meaning. Ideally, the closer two word embeddings are, the more they represent close concepts. The distance between word embeddings may be computed, for instance, using *cosine similarity* (normalized dot product between both vectors), which can be written as follows:

$$\cos(t, e) = \frac{te}{\|t\|\|e\|} = \frac{\sum_{i=1}^n t_i e_i}{\sqrt{\sum_{i=1}^n (t_i)^2} \sqrt{\sum_{i=1}^n (e_i)^2}} \quad (1.1)$$

with t and e defining the word embeddings.

1.2.1 Non-contextual embeddings

The following models generate non-contextual embeddings, i.e. the vector representations are the same for each word whatever the context. While the embeddings do not take into account the context, they are often built using words and their context in order to learn the best possible embeddings. These works are all based on the distributional hypothesis (Widdowson, 2007) according to which words that occur in similar contexts tend to have similar meanings. They follow the founding work of Bengio et al., 2003, which, for the first time, has proposed to associate each word with a vector representation. It also introduces the principle of similarity of representations if words and contexts are similar. However, the context of a word has a fixed size as highlighted by Mikolov et al., 2013b with Word2Vec.

Word2Vec

Word representations are learned from the word's context to allow for a better understanding of the text. Word2Vec (Mikolov et al., 2013b) models are composed of two neuron layers. These models map words onto a multi-dimensional vector space. Each word corresponds to a vector in this space. The dimension is usually a few hundreds but is varying depending on the model. Word2Vec models are designed for self-supervised training. Self-supervised training is a supervised training where data is not labeled by humans but found in the original data. For example, masked word prediction is a self supervised technique.

Two variants of Word2Vec were introduced (CBOW (Continuous Bags Of Words) and Skip-Gram (see Figure 1.2)).

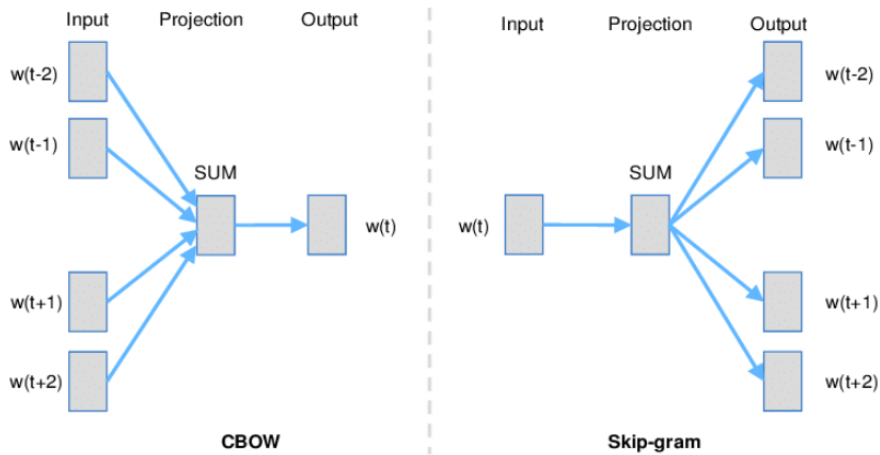


FIGURE 1.2: Word2Vec (Mikolov et al., 2013b) models

The CBOW model learns the embeddings by predicting a target word from its context, i.e. the words next to it in the sentence (leftside of Figure 1.2). The model is composed of two neuron layers. The first one maps the context (ids of the words in the vocabulary) onto the representation space using the sum of the context's word representations. Considering the size k of the context and the index i of the targeted word, this representation can be computed as follows:

$$R_{context} = \sum_{w=i-k}^{i+k} MO_w \quad (1.2)$$

with $R_{context}$ the vector representation of the context, M the matrix containing the word representations and O_w the one-hot vector representing the word w of the sequence. A one-hot vector is simply formed from the index of the word in the dictionary:

$$O(w) = (e_{w,1}, e_{w,2}, \dots, e_{w,n}) \quad (1.3)$$

with $e_{i,j} = 1$ if $i = j$ and 0 otherwise, and n the size of the vocabulary. The representation of the context is therefore dense compared to a sparse representation obtained with a simple Bag of Words, hence the name *continuous*. The second neuron layer helps predicting the output distribution from the context representation. This distribution is computed over all possible words in the vocabulary using a softmax function. This layer is only used during the training because the purpose of the model is not to predict words given the context but to learn rich representations for

each word. The model is trained using text from books or articles. When the training has ended, only the internal representation matrix matters. It contains the word embeddings on its rows, with row i corresponding to the representation of word i in the vocabulary. CBOV model is powerful for when it comes to learning representations of frequent words and is fast to train.

The Skip-Gram model learns the representations by predicting the context given a target word (right side of Figure 1.2). It works well for limited datasets and rare words. Indeed, it is easier to predict the context of a rare word than predict the rare word given the context.

The richer the vocabulary is, the more computations are required. That is why several methods were developed to make the computations less numerous or faster. To obtain the probability of a target word or a context word, the softmax on all possible words must be computed (N computations). To save time, it is possible to transform the last transition matrix into a decision tree which reduces the number of computations to m with $N = 2^m$. This method is called *Hierarchical Softmax* (Morin and Bengio, 2005).

In order to reduce the number of computations, it is also possible to train the model not to output the wrong words instead of outputting the right words. This is called *Negative Sampling* (Mikolov et al., 2013a).

GloVe

GloVe (Pennington, Socher, and Manning, 2014) is an algorithm based on the same principle as Word2Vec (Mikolov et al., 2013b). It is able to learn word embeddings using the context of the words but it also takes in account some global statistical information like the words frequency and the co-occurrences to improve the meaning of the word representations.

fastText

fastText (Bojanowski et al., 2016) is a Facebook library for efficient learning of word representations and sentence classification. Each word is represented by its sub-words. For instance, artificial=ar,art,rti,tif... The size of the window is chosen by the user. This turns out to be helpful to capture the meaning of suffixes and prefixes. The words are first transformed into sub-words and their representation is then learned using Skip-Gram model.

fastText (Bojanowski et al., 2016) is therefore an improvement of Word2Vec (Mikolov et al., 2013b) model using sub-words information. FastText works well for rare words.

Discussion

Word2Vec (Mikolov et al., 2013b), GloVe (Pennington, Socher, and Manning, 2014) and fastText (Bojanowski et al., 2016) are word embedding models which are trained using words' contexts. They are able to model syntactic and semantic information. For example, an interesting property of Word2Vec models is additivity. For example $R(\text{"Queen"}) = R(\text{"King"}) - R(\text{"Man"}) + R(\text{"Woman"})$ with $R(\text{"w"})$ the representation of w . Nevertheless, in production, the context is no longer taken into account and the words do have static representations. It means that a same word in two different contexts has the same representation (for example *fly* in "I will *fly* a plane" and in "I see a buzzing *fly*"). The embedding process can, therefore, still be improved, especially through contextual embeddings.

1.2.2 Contextual embeddings

Contextual embeddings are usually generated using deep neural encoders. Several architectures are used in language processing such as recurrent networks, transformers or more rarely convolutional networks.

Recurrent Neural Networks (RNN)

Recurrent networks are characterized by a temporal link between the network outputs h_t and inputs x_t (see Figure 1.3). Indeed, hidden states (the network's memory cells) are updated with respect to the information flowing from the input to the output as follows:

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \quad (1.4)$$

with h_t the hidden states at time-step t , W_{ij} parameters of the model and x_t the input at time-step t . The output y_t is then computed with:

$$y_t = W_{hy}h_t. \quad (1.5)$$

Some information are therefore "saved" and can be reused later, i.e. for future predictions. This temporal link makes the recurrent networks particularly relevant for Natural Language Processing tasks as they are links between words that appear at different time steps and positions. Nevertheless a major drawback of vanilla recurrent networks was vanishing or exploding gradient (Pascanu, Mikolov, and Bengio, 2013). The weights are indeed involved in the back-propagation for each time step which, depending on its largest singular value s , implies a vanishing gradient ($s < 1$) or an exploding gradient ($s > 1$).

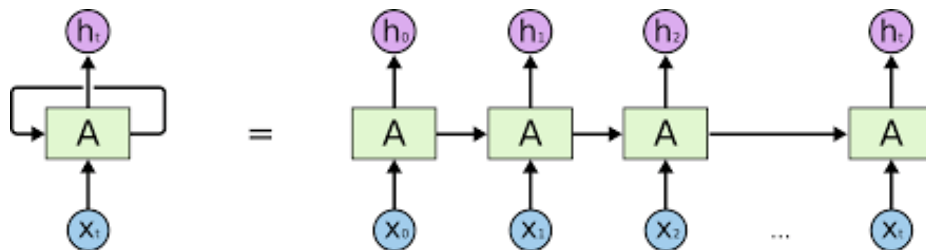


FIGURE 1.3: Structure of a recurrent network

Another structure of recurrent network was thus introduced to avoid the convergence problems, the Long Short Term Memory (LSTM). The internal structure of an LSTM (see Figure 1.4) contains several "gates" which control the information in the hidden states. The memory vector flows in the upper part of the cell in Figure 1.4. The network input x_t and the previous network output h_{t-1} flow in the lower part in the cell. The hidden states may be modified in two particular areas. The first one is the "Forget Gate Layer". A Hadamard product of the hidden states and a sigmoid function depending on the input and previous output of the cell is performed. This gate decides whether the saved information is still of importance or not. New information may then be added to the saved states using the second gate, called input gate. The hidden states are therefore updated for every prediction the network makes. The consequence on the back-propagation is that the gates actually control the gradients and always keep it small through the use of sigmoid function to add or remove information from previous states. Moreover, while the same parameters were used at each time step to compute the cell states, in the vanilla RNN, the cell

state is computed through the forget gate. The input gate just copies inputs to the cell state and, therefore, gradients are only copied in the back-propagation process.

In the case of a text analysis, the words are successively fed to the network. It means that the network is only able to analyze the previous words to find the context. A LSTM network is therefore unidirectional. It works either from left to right or from right to left but it cannot analyze both right and left contexts. To address this problem, two LSTM networks may be stacked: the first one processes the context from left to right and the other one from right to left.

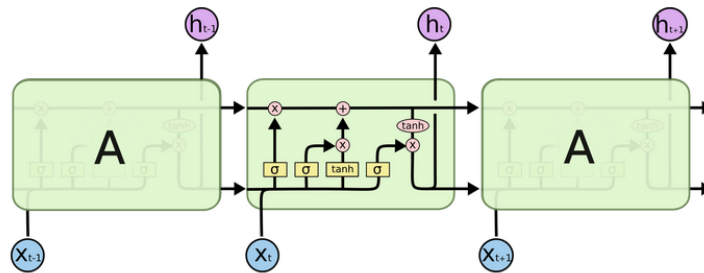


FIGURE 1.4: Structure d'un réseau Long Short Term Memory

SummaRuNNer (Nallapati, Zhai, and Zhou, 2017) and NeuSum (Zhou et al., 2018) are GRU-based networks for extractive summarization. A GRU network is very similar to a LSTM. The main difference is the fusion between the hidden states and the output in a GRU network, i.e. the outputs are the hidden states. In a LSTM, the outputs are different from the hidden states.

Recurrent Networks show good performance for text analysis and text summarization (Zhou et al., 2018; Nallapati, Zhai, and Zhou, 2017). It works also very well for generation tasks as it is a directional task. The words are predicted one after the other. Nevertheless the main limitation of recurrent networks is the flow of information. To analyze a sentence, the network updates the hidden states at each time step, i.e. for each word. It means that the information coming from the first words is drowned under the other words information. Therefore, the quality of the transferred information decreases relatively fast. Research was therefore focused on designing a fully bidirectional network instead of stacking two unidirectional networks to build a bidirectional network. This led to the development of transformers and pre-trained encoders.

Transformers

Encoders may be designed to be trainable in a non-supervised manner. This is especially the case for transformer networks (Vaswani et al., 2017). They can be trained on all sort of documents like Wikipedia pages, articles, journals. These models are largely used due to the lack of labeled data in many tasks. With a small dataset, it is difficult to train the encoder and the decoder in the same time without over-fitting and under-performance.

If the encoder is already pre-trained on another dataset, then a small dataset may be enough to learn a downstream task without overfitting. These encoders are therefore trained on very large datasets (billions of words) and their performances are beyond reach. BERT (Devlin et al., 2019) (Figure 1.5) is the most famous pre-trained encoder. Transformer networks are interesting for language understanding and language generation because of their bidirectionality. They can also be parallelized easily which makes them faster than recurrent networks. They are composed

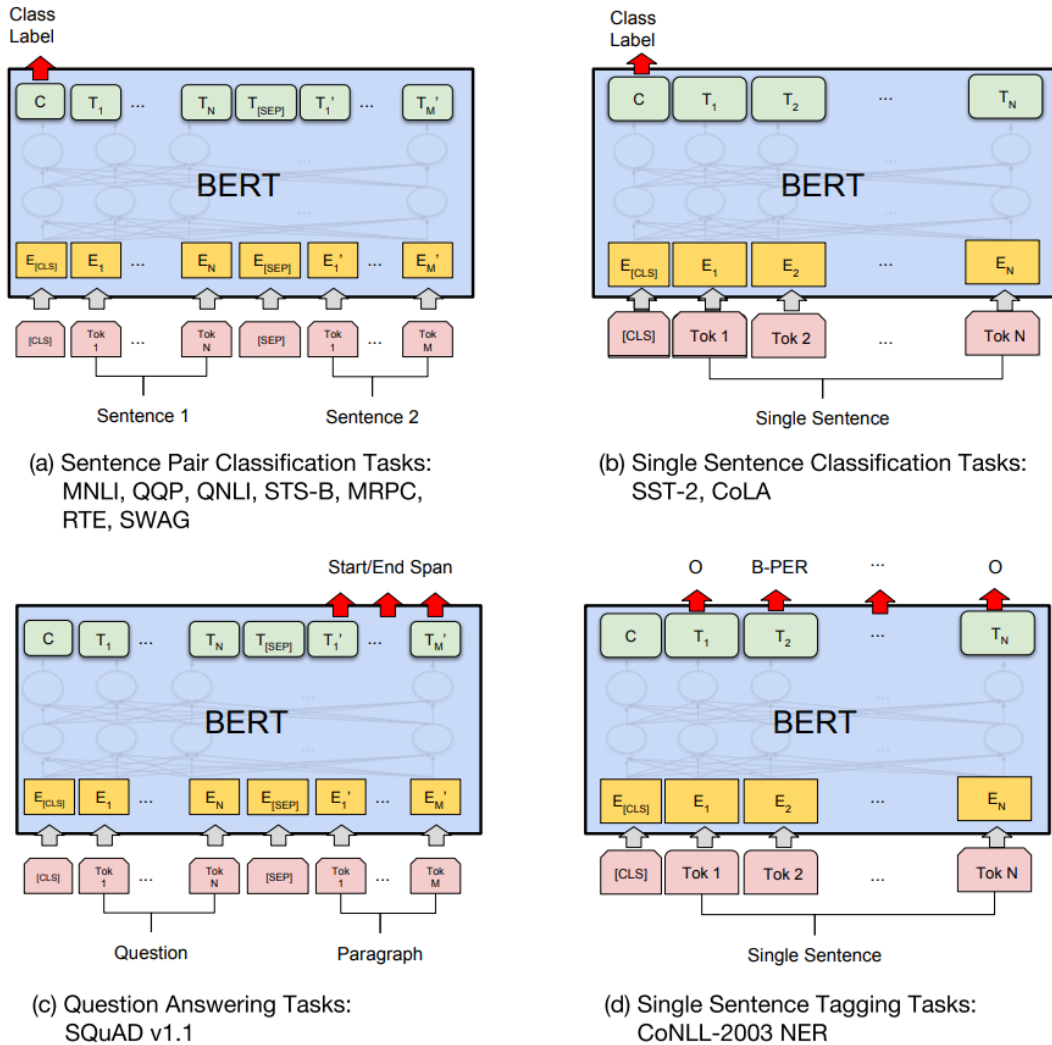


FIGURE 1.5: BERTDevlin et al., 2019 specialization for downstream tasks

of several stacked *Attention Mechanisms*. An Attention Mechanism is designed to select the best information from a flow of data. For instance, it can be used to determine which encoded information is important for the decoder. There are two different Attention Mechanisms: global attention, where all the information are weighted (according to their relevance) and summed up, and local attention, where only a part of the information is taken into account to reduce the number of computations.

The stacked attention mechanisms in BERT (Devlin et al., 2019) are composed of three inputs (see Figure 1.6): the Keys (K), Queries (Q) and Values (V). The Queries represent the information needed by the network. The Keys are the actual information flowing through the network. A dot-product between the Queries and Keys therefore highlights the information which should be kept from the Values with a simple multiplication. Indeed, Values do also contain the flowing information in the network. When the Keys, Values and Queries contain the same information, it is called self-Attention. These mechanisms enable the network to study the internal structure of a text:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (1.6)$$

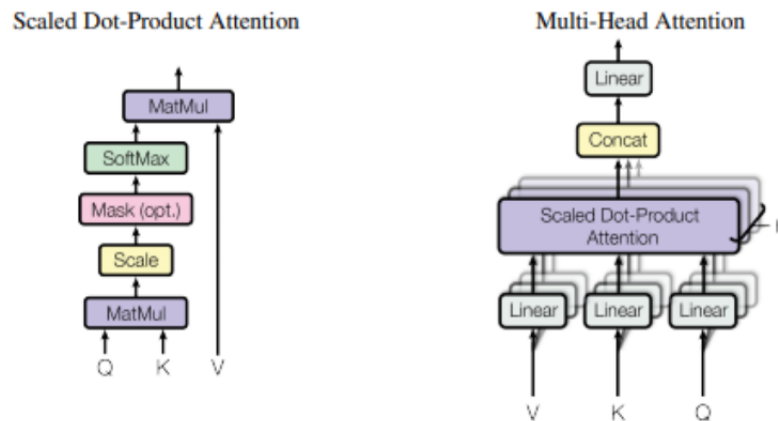


FIGURE 1.6: BERT Devlin et al., 2019's Attention Mechanism

with d a rescale factor linked to the dimension of the input. For instance, on the example of Figure 1.7, the attention values show that there is a strong connection between "The", "animal", "cross" and "street". It actually highlights all the words linked to the animal.

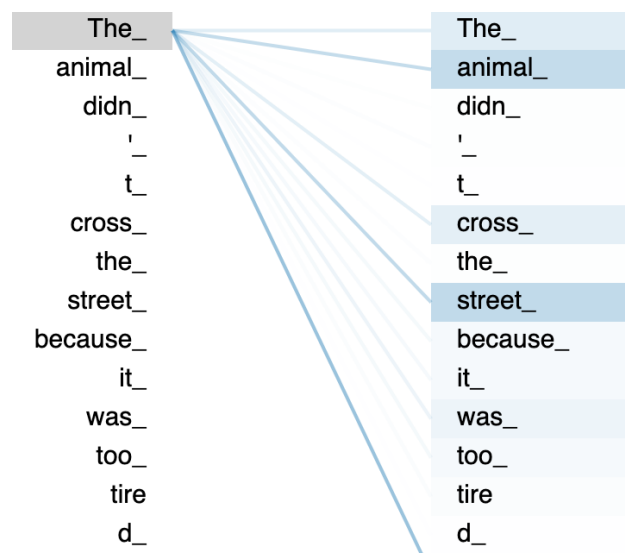


FIGURE 1.7: Self attention example

BERT encoder shows incredible results and has established the state of the art in a great number of NLP tasks. Nevertheless, it has a few drawbacks. The self-supervised training is based on the prediction of masked input tokens. To predict one of these tokens, the model should rely on all the other tokens but some of these are also masked. Moreover, it does not take into account the dependencies between masked tokens. The input length is also limited to 512 tokens because of the chosen position embedding for tokens, and to limit the number of parameters.

In Lan et al., 2019, several sharing parameters and learning techniques are presented in order to lighten BERT model and improve its performance. Unfortunately, the global limitations of BERT cited above are not solved.

XLNet (Yang et al., 2019a) model aims at addressing the limitations of BERT approach and the classical auto-regressive limits by mixing the advantages of both techniques: the ability of generation of the auto-regressive models (e.g. recurrent

models) and the pre-training and bidirectionality of BERT without its learning issues. XLNet is based on a permutation training. The context words are randomly shuffled and the network is trained to predict each word given the previous words, that can come from all over the sentence. The model is therefore able to gather information from all positions on both sides. XLNet performs better than BERT in various NLP tasks but mostly because it was learned using much more data than BERT, as shown in Zhuang et al., 2021 where the authors compare XLNet and BERT performances with an equal amount of training data.

Other self-supervised transformers were also developed. (Lewis et al., 2020b) introduced BART. BART is an encoder-decoder model which was trained to denoise input sequences. A noisy sequence is passed to the encoder and the decoder must predict the original sequence. Five types of noising methods are used: token masking, sentence permutation, document rotation, token deletion and text infilling. It outperforms BERT in several tasks including abstractive summarization.

Convolutional Neural Networks (CNN)

Convolutional networks (Lecun et al., 1998) are derived from image processing and are more rarely used in NLP than the two types of networks discussed above. Nevertheless, they have properties that can also be exploited for language processing.

CNNs are composed of successive filters that extract features from the input data. In images, videos or texts, these filters detect spatial and temporal dependencies between the features. The convolution mechanism is illustrated in Figure 1.8 with an RGB image. The filter, also called kernel, has a certain dimension (3×3 in this example) and if the image has 3 channels (R, G, B), it will also be the case for the kernel (which is equivalent in this case to having a kernel of dimension $3 \times 3 \times 3$). The kernel is then applied and moved on the image from left to right and from bottom to top so that a new image of the same dimension is created in output. This operation mathematically corresponds to a cross-correlation. Each element of the output image corresponds to the result of applying the filter at a certain position. For a given position, the elements of the kernel and the image are multiplied one by one and then summed. A bias can be added to the result. The convolution on a gray-scale image can be described with the following equations:

$$z_{ij}^l = \sum_{a=0}^m \sum_{b=0}^{m-1} w_{ab} y_{i-\frac{m-1}{2}+a, j-\frac{m-1}{2}+b}^{l-1} \quad (1.7)$$

$$y_{ij}^l = \sigma(z_{ij}^l) \quad (1.8)$$

with z_{ij}^l the pre-activation for pixel (i, j) in layer l , m the size of the filter, (a, b) the position of the weight w_{ab} in the filter and y_{ij}^{l-1} the output at position (i, j) of layer $l - 1$. σ is the activation function of layer l .

To ensure that the output image keeps the same dimensions as the input image, arbitrary values can be added to the edge of the input image (here 0's). This operation is called padding.

The convolutions work in the same way for texts (Amplayo et al., 2018; Mou et al., 2015), the only difference being that there is usually only one dimension in input instead of 3 for images. CNNs can also be associated with encoders of different architecture as in Zhou et al., 2016 with an LSTM type encoder. CNNs have a strong capacity to extract features locally but are limited when it comes to creating longer

term dependencies, between several sentences for instance. That is why they remain much less used in language processing so far. Nevertheless, Li et al., 2021 have recently shown that it is possible to implement attention mechanisms with convolutional networks.

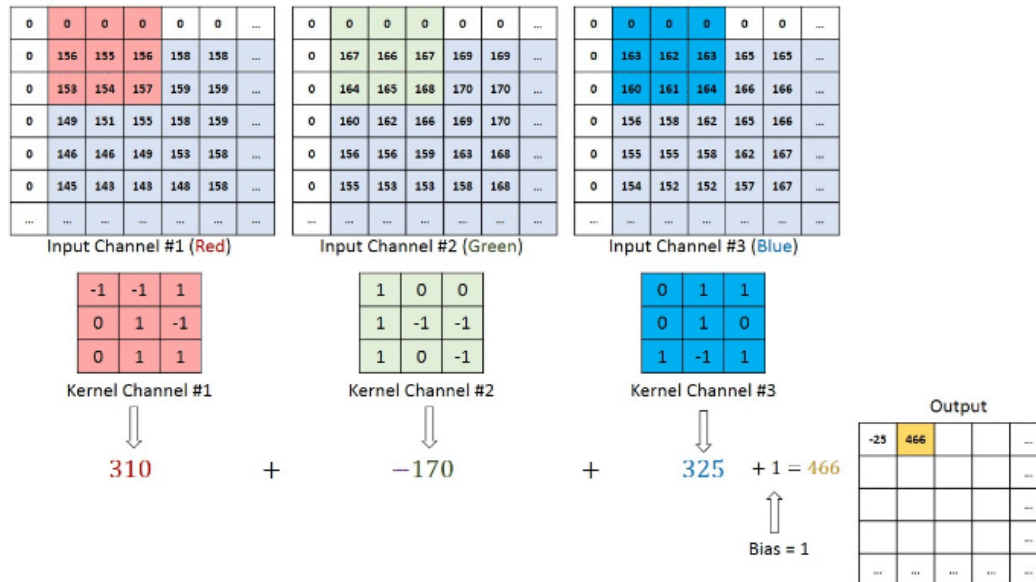


FIGURE 1.8: Convolution in a CNN

1.3 Targeted content identification and extraction

Our objective is the extraction of textual elements allowing us to characterize problems and solutions. Depending on the level at which we place ourselves (word, sentence, document), several different processes are possible. For a word classification, Named Entity Recognition is the closest task. For a sentence classification, the automatic summarization will be chosen. A document classification can also be studied to filter the targeted documents. Finally, generative models can also be exploited to identify, copy/reformulate textual elements of interest. A classifier is necessary for any classification task. A classifier is an algorithm that predicts the class to which a piece of data belongs, i.e. characterize the data. Several types of classifiers exist, some of them require prior learning (Eager learners) while others delay the processing of training datasets until it receives the test instance for the class prediction (Lazy learners). For example the KNN algorithm (Fix and Hodges, 1989) (allowing Case-Based Reasoning) consists in using a dataset of pairs (x, y) with x the input (for example an embedding) and y the output (label). When an input data X is provided to the model, the latter looks for the k data (x_i) in the dataset which are the closest to X with a distance criterion corresponding to the nature of X (vector, float etc...). The class Y , the most represented in the k selected data, will then be associated to X . The KNN is a lazy learner because the dataset is not used until inference time. On the contrary, a perceptron or Multi-Layer Perceptron (MLP) (more details about MLP fundamentals are provided in Appendix B) requires a training prior to the inference. They are eager learners. Decision trees, logistic regressors or Support Vector Machines (SVM) are also part of this category. Nevertheless, they all have the same

drawback. Each prediction is independent of the previous predictions. Creating dependencies in the predictions is essential when processing sequential data like texts. The Conditional Random Field (CRF) was introduced for that purpose. The CRF is able to model dependencies between prediction and bring consistency in a sequence classification.

1.3.1 Document classification

The embeddings built by the state of the art models are token embeddings. In order to classify documents, it is necessary to build document representations from token representations. One of the methods traditionally used is to make an average of the token representations. This technique is used by Le and Mikolov, 2014 which present Doc2Vec, an approach based on word-wise representations from Word2Vec (Collobert et al., 2010).

Some encoders can be pre-trained to directly generate document representations. This is the case of BERT which has been trained on a *Next Sentence Prediction* task. This task consists in predicting whether the two input sentences have a logical relationship. For this purpose, a special classification token is introduced, which can be reused for other classification tasks. However, most of the pre-trained models do not have a classification token and the token-wise embeddings have to be exploited.

1.3.2 Named Entity Recognition (NER)

Named Entity Recognition (NER) is a task consisting in identifying Named Entities like people, places, company names, etc., in textual contents. It is a very common task in NLP since it allows identifying keywords.

NER was introduced in 2003 with the release of Conll dataset. The first models were based on rules and lexicons but they were quickly replaced with Maximum entropy models (Bender, Och, and Ney, 2003), Hidden Markov Models (Florian et al., 2003, Klein et al., 2003) and perceptrons (Carreras, Màrquez, and Padró, 2003). The Long Short Term Memory networks (LSTM) were particularly suited for this task (Hammerton, 2003a).

NER models then shifted towards word embedding models based on Word2Vec (Collobert et al., 2010) or recurrent networks (Habibi et al., 2017). NER models can be built from word or character embedding models. Ma and Hovy, 2016 proposed a word and character-based model.

Latest models are based on pre-trained transformers as they are able to build richer token representations (Li, Zhang, and Zhou, 2020; Yu et al., 2019; Xu et al., 2021).

Named Entity Recognition may be used as a first step in a relationship extraction process. It consists in identifying relations between words or entities and can be considered as a (multi-label) classification task (Allahyari et al., 2017).

1.3.3 Summarization through sentence classification

Extractive summarization techniques consist in selecting the best sentences regarding the summarization objective, in our case retrieving the sentences introducing contradictions. An extractive summarization enables the retrieval of unbiased information but the coherence of the summary may be low due to a stitching of the summary sentences without real links between those.

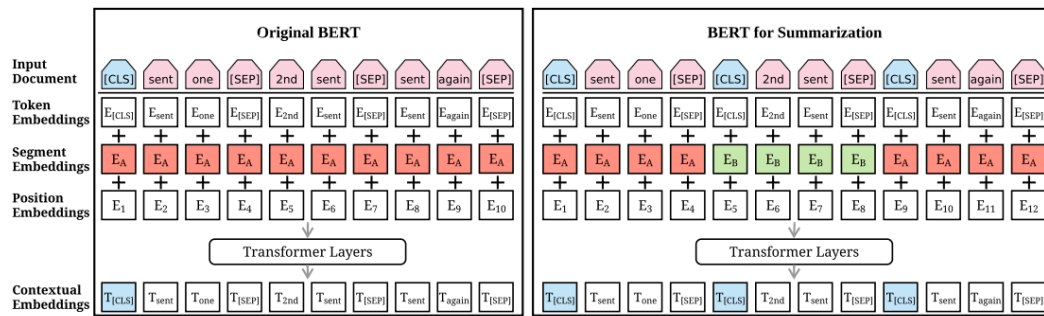


FIGURE 1.9: Summarization network

The first summarization algorithms were based on graphs (Page et al., 1998; Kleinberg, 1999; Mihalcea, 2004; Litvak and Last, 2008). Nodes (or vertices) of a graph representation are objects (or entities) and the interconnections between these objects are edges (or links). The nodes can be words or concepts and the edges can be for instance a semantic distance or coreferences for web-based applications. This approach has been used in PageRank (Page et al., 1998) or HITS (Kleinberg, 1999). Nevertheless, the accuracy of such algorithms remains limited. Naive bayesian approaches (Kupiec, Pedersen, and Chen, 1995; Aone et al., 1997), Hidden Markov models (Conroy and O’leary, 2001) or Conditional Random Field-based models (Shen et al., 2007) were also proposed. Al Saied, Dugué, and Lamirel, 2018 show a parameter-free approach for extractive summarization using Feature Maximisation (Lamirel et al., 2004).

With the emergence of deep learning, most neural-based extractive summarization algorithms consist in recurrent networks. Recurrent networks are able to catch time dependencies which makes them very suitable for NLP applications. LSTMs (Long Short Term Memory networks) or GRU (Gated Recurrent Unit) were especially used. Recurrent networks are by nature unidirectional. They can catch time dependencies along one direction. This problem was addressed with the introduction of stacked recurrent networks to catch dependencies in both directions (SummaRuNNer (Nallapati, Zhai, and Zhou, 2017)) and NeuSum (Zhou et al., 2018)). Reinforcement learning was also studied (Narayan, Cohen, and Lapata, 2018b) but these algorithms only show their full potential in abstractive summarization.

At present, the most widely used extractive summarization networks are built with pre-trained encoders like BERT (Devlin et al., 2019) or XLNet (Yang et al., 2019a). New research directions have been proposed to see the problem of summarization at another level than the traditional sentence level. Summarization can for instance be considered as a matching problem at the summary-level (Zhong et al., 2020). Nevertheless, as the number of sentences to extract from the patents is unknown, these approaches cannot be used.

1.3.4 Summarization through generation

Sequence models are able to generate a textual content from another textual content.

These models can be neural networks but this is not always the case. For example, in the case of automatic summarization, Oya et al., 2014 propose a model based on templates of existing summaries. Depending on the similarity of a document with one of the templates, the summary is built on it and consists in filling in the

template gaps. We can also cite Gatt and Reiter, 2009 which presents a tool able to generate text from syntactic rules.

The architecture of a Seq2Seq neural network consists of a decoder on top of an encoder. The encoder will extract the important features for the task at hand and encode each token as a vector of fixed size. For example, in the case of an automatic summary, the encoder will identify the important information for the summary. The decoder is responsible for generating correct sentences with respect to the encoded data. A decoder recursively generates tokens. At each step, given the previous tokens and the encoded data, it generates the token that statistically has the highest chance of appearing. These models are called *language models*.

The abstractive summarization aims at generating a new text summarizing the input document. The advantage is that the readability is high since the sentences are theoretically linked, but this can lead to misinterpretation if the model modifies the sentences too much, which can be a problem for retrieving reliable information.

State of the art abstractive summarization models (Song et al., 2019; Zhang et al., 2019) are based on deep neural networks, mostly pre-trained transformers with an encoder-decoder architecture. pre-trained encoders like BERT (Devlin et al., 2019) or XLNet (Yang et al., 2019a) are also used for abstractive summaries. A transformer network with the same structure can be stacked on top of the encoder to generate the summary (see Figure 1.10).

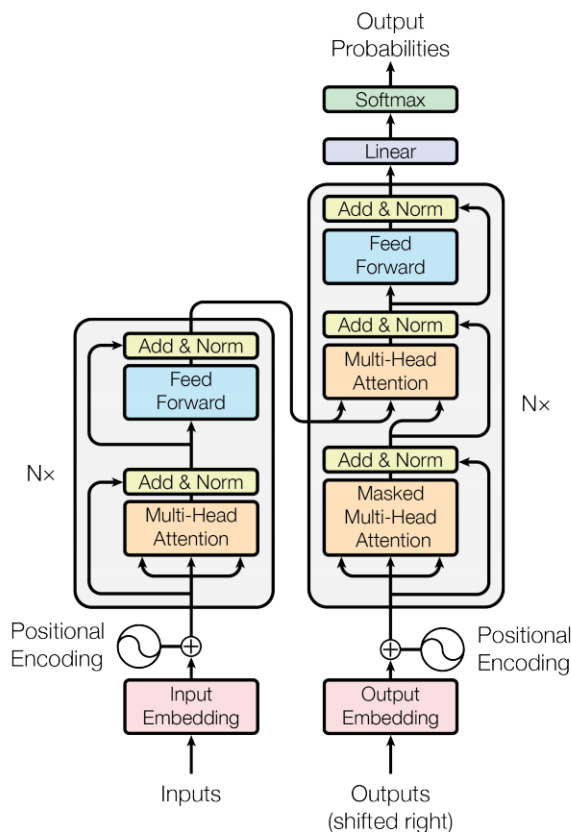


FIGURE 1.10: Decoder with BERT (Devlin et al., 2019) encoder

The output of the encoder is fed into each decoder's block. The decoder analyzes the previous output to determine what kind of data it needs. Then it selects the data coming from the encoder through a multi-head attention. Finally, it computes a

probability distribution over the entire vocabulary whose maximum indicates the next output word.

The performance of pre-trained models is studied in (Raffel et al., 2019). It introduces a unified framework to see each NLP task as a text-to-text task. Some examples are shown on Figure 1.11. It investigates the impact of pre-training objectives, architectures and number of parameters, unlabeled datasets, and transfer learning approaches.

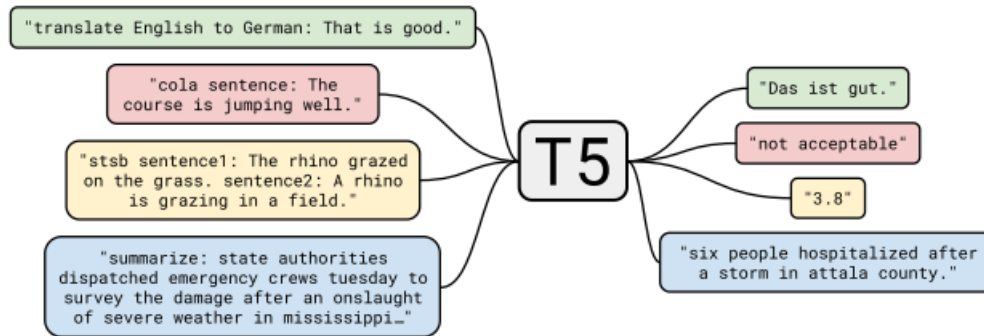


FIGURE 1.11: Unified text-to-text framework (T5)

Unified Language Model (UNILM)(Dong et al., 2019) was developed to improve BERT performance for language understanding but also for language generation. BERT pre-training is based on bidirectional context but language generation tasks are by nature, unidirectional. That is why adding this possibility may improve BERT language generation capabilities. For UNILM, three kinds of pre-training may be used: bidirectional pre-training (where both the right and left context words are given), left-to-right unidirectional pre-training (all the words on the left of the target word are given) and right-to-left unidirectional pre-training (all the words on the right of the target word are given). This leads to better results than BERT's for language generation tasks and especially for abstractive summarization but not for extractive summarization. MASS (Song et al., 2019) is another pre-trained model in which the decoder is also pre-trained. Different words are masked to learn the encoder and the decoder in the same time. The risk of overfitting of the decoder when trained from little databases is then lowered. The purpose of the PEGASUS (Zhang et al., 2019) model is to adapt the pre-training directly to the abstractive summaries. In this case, entire sentences are predicted by the model during training. Therefore, the model can learn to build coherent sentences before fine-tuning with a summary database. The results are better than other techniques for abstractive summaries.

(See, Liu, and Manning, 2017)(Nallapati et al., 2016) introduce a LSTM-based pointer-generator model with mechanisms to avoid reproducing inaccurate details and repeating sequences in the summary. The models are based on LSTM networks for both encoder and decoder. An attention mechanism is used on top of the encoder to select which part of the information in the encoder is sent to the decoder. This information and the output of the decoder are then used to choose if an input word is copied or not. A coverage mechanism is also utilized to keep track of which part of the documents were already used for the summary. Therefore, repetitions are more unlikely.

In (Wang et al., 2019), an extractive model is used with an abstractive model to jointly select and rephrase the sentences from the original text. The extractive and abstractive models are pre-trained and then fused in a reinforcement learning framework. The entire model is therefore trainable end to end.

Convolutional Neural Networks (CNN) may also be used for summary generation (Narayan, Cohen, and Lapata, 2018a). They try to capture long range dependencies using stacks of convolutional masks. The network is topic-conditioned, each word is transformed into a topic-vector using a convolutional encoder. The convolutional decoder takes these topic-vectors as input and selects the most important ones.

Generative adversarial networks can be used for abstractive summarization. A generator is trained to output a summary from a random text and the discriminator tries to distinguish the generated summary from the golden summary (Xu et al., 2018)(Liu et al., 2017). The summaries are coherent but their ROUGE score (a summary quality metric) is not as high as the leading methods from the state of the art. The fact that the ROUGE measure is biased must be taken into account as it only counts similarities between the output summary and the gold summary.

Lately, the giant networks based on transformers with several hundreds of billions of parameters such as GPT-3 (Brown et al., 2020), Megatron (Smith et al., 2022) or BLOOM are able to do automatic summarization with a few examples (using few-shot learning) but their use is complicated because of their size.

Conclusion

In this chapter we have presented the most interesting approaches for text mining and information retrieval. Supervised neural approaches are pointed out as the most performing models, considering the state of the art results on data mining tasks (summarization, NER, ...). Pre-trained transformers (Devlin et al., 2019; Yang et al., 2019a; Dong et al., 2019) have the advantage of requiring very little annotated data thanks to the quality of the generated token-wise representations. In this thesis, we seek to identify key concepts in patents that describe inventions. These concepts are detailed in the next chapter. The NLP task which would allow performing this concept identification is Named Entity Recognition which is in fact a token-wise classification. Nevertheless, to identify key phrases, automatic summarization is also useful and is exploited in the rest of the thesis. The summary, extractive or abstractive, has different characteristics (fidelity, readability, etc.) which will also be studied.

Chapter 2

TRIZ and Inventive Design in patent documents

Contents

Introduction	21
2.1 TRIZ theory	21
2.2 Patent Mining	24
2.2.1 Patent structure	24
2.2.2 Available datasets	25
2.2.3 Patent search	26
2.2.4 Patent classification	27
2.2.5 Generation tasks	27
2.3 TRIZ and patent mining	27
2.3.1 Contradictions and TRIZ parameters extraction	28
2.3.2 Inventive principles, physical effects and solution mining	29
2.3.3 Estimation of inventiveness level	30
Summary and conclusion	30

Introduction

In this chapter the TRIZ theory (Altshuller, 1984) and the Inventive Design methodology (Cavallucci, 2012), both exploited in this work, are presented. Inventive Design Methodology (IDM) is built on top of TRIZ theory to provide definitions to all key concepts. The concepts of parameters and physical/technical contradictions in TRIZ theory suffer from poor definitions which lead to divergent interpretations. The definitions from IDM are used to structure the algorithmic approach to TRIZ key concept mining.

All the key concepts from TRIZ/IDM, especially the contradictions, are defined in this chapter. Special attention is given to the works combining TRIZ theory and patent mining.

2.1 TRIZ theory

The TRIZ theory was developed by Genrich Altshuller (Altshuller, 1984). Altshuller is known for his research focusing on formalizing the invention mechanism in the form of a repeatable algorithm. Based on observations of inventions described in

patents, Altshuller found that the processes leading to inventive solutions of complex problems were similar in all fields. He therefore worked on general formulations of the concepts of problems and solutions. One of his major contributions was the definition of the notion of contradiction. Let us take the example of an airplane whose interior volume needs to be increased. It is then necessary to increase the size of the structure of the plane. However, if the size of the structure increases, the weight of the plane increases as well, which is disadvantageous considering fuel consumption. Altshuller proposes to formulate all problems as a contradiction. In this example, there is a contradiction between volume and weight since when the volume parameter is improved (here, by increasing it), the weight parameter is degraded (in this case, it increases). To be able to compare problems from different domains, Altshuller introduced 39 parameters (commonly called TRIZ parameters, Table 2.3). The contradictions between these parameters are able, in theory, to describe any problem.

Altshuller was also interested in the solutioning. So, as with the parameters, he tried to find universal formulations of the ideas leading to solutions. This is what he called the inventive principles. There are 40 of them (Table 2.4). The contradictions and the inventive principles allowed him to develop the TRIZ matrix and the inventive problem solving algorithm presented in Figure 2.1. The TRIZ matrix is a statistical representation of the inventive principles that were mostly used in the 40,000 inventive patents that were analyzed by Altshuller and the engineers with whom he worked with. Each cell corresponds to a contradiction between two evaluation parameters (which are on the abscissa and ordinate). The cells contain, for each contradiction, the indices of the inventive principles that were statistically the most used to solve these contradictions. The process of inventive solving (Figure 2.1) exploits the structure of this matrix. A problem is formulated as a contradiction by using specific parameters. These parameters are then translated (by similarity) to obtain a contradiction between two TRIZ parameters (among the 39 original parameters). The cell associated to this contradiction gives the concepts to be studied in the form of inventive principles. These must finally be interpreted considering the studied system to solve the problem. This method has proven its worth and is still very popular in Asia.

Inventive Design Methodology (IDM) is a theory introduced in (Cavallucci, 2012), built on top of TRIZ theory and which, through the construction of a real ontology, endorses some definitions such as that of contradiction and parameters. IDM is not the only existing ontology related to the description of objects or solution concepts. SAO (Subject Action Object) can also be applied (Kim, Park, and Yoon, 2020).

In this thesis, we use all the concepts defined in IDM. The contradiction in the original TRIZ theory is only characterized by the two contradiction parameters. In IDM, these two parameters are called Evaluation Parameters (EP) while an additional parameter, called action parameter (AP) is introduced. The action parameters are modifiable parameters of the studied system having an influence on the evaluation parameters. The evaluation parameters are described as the parameters of the system allowing to measure the positive or negative influence of the modification of the action parameters on the system. For instance, in the example of the aircraft used previously, the action parameter (AP) is the size of the structure. This one has a positive influence on the volume (EP) and a negative influence on the weight (EP) when it is increased, and conversely when it is decreased. Partial solutions are introduced to represent the non-ideal solutions. It is strongly linked to the action parameters. A partial solution, through an action parameter, is able to improve an evaluation parameter but leads to the degradation of another evaluation parameter.

Num.	Separation principles
1	Separation of contradictory modalities in space.
2	Separation in time.
3	Combination of several systems: "polysystem".
4	Combination of a system and its opposite: "anti-system".
5	Separation between a system and its subsystems: the system has property A while the subsystems have property B.
6	Transition to the "micro-level": change of scale by the use of substances in a more "dissociated" physical state a more "dissociated" physical state: powder, liquid, gas....
7	Phase change of a part of the system, or of its environment.
8	Dynamic" phase change depending on the working conditions (phase change in time).
9	Use of the phenomena associated with phase changes.
10	Replacement of a single-phase substance by a bi- or polyphase substance.
11	Creation/elimination of substances by physical-chemical combination or decomposition.

TABLE 2.1: Separation principles

Two types of contradictions can be defined: technical and physical contradictions. These two types of contradictions are not always related to different problems but just to different formulations of the problems. The technical contradictions emphasize the evaluation parameters: "it is impossible to satisfy these two EPs at the same time" (for example weight and volume). The physical contradictions, on the other hand, focus on the action parameter: "it is impossible for this AP to be at the same time *state 1* and *state 2*" (for example small and large) "so that both EPs are satisfied at the same time".

The matrix can be used to solve technical contradictions. Physical contradictions cannot be solved directly due to the contradictory nature of the action parameter. To achieve this purpose, the contradiction must be "broken" with a separation principle. These separation principles are presented in Table 2.1. Let us take the example of a washbasin, it must be both low for children and high for adults. However, we can hypothesize that adults and children do not use it at the same time (principle 2) and that a solution would therefore be to have a vertically mobile sink. Principle 3 could also be exploited and have several sinks at different heights.

Several levels of inventiveness are used to describe the found solutions. Altshuller defined 5 different levels of inventiveness (Altshuller, 1984) which range from an obvious solution (Level 1) to a discovery (Level 5). The more the idea of the solution comes from a distant domain, the higher the level of inventiveness. The different levels of inventiveness are listed in Table 2.2. Level 1 does not correspond to an inventive resolution as the key element of the resolution is already known in the domain. On the contrary, levels 2 and 3 are considered inventive. According to Altshuller, more than 75% of the patents are at levels 1 and 2. Levels 4 and 5 are much rarer as they call upon remote or still unknown knowledge (Level 5) to solve the contradiction.

Level	Definition
1	Simple improvement of a technical system. Requires knowledge available within an industry relevant to that system.
2	Resolution of a technical contradiction. Requires knowledge from different areas within an industry relevant to the system.
3	Resolution of a physical contradiction. Requires knowledge from other industries.
4	Development of a new technology. It is developed by using breakthrough solutions which require knowledge from different fields of science.
5	Involves the discovery of new phenomena. The new phenomenon is discovered that allows pushing the existing technology to a higher level.

TABLE 2.2: Inventivity Levels

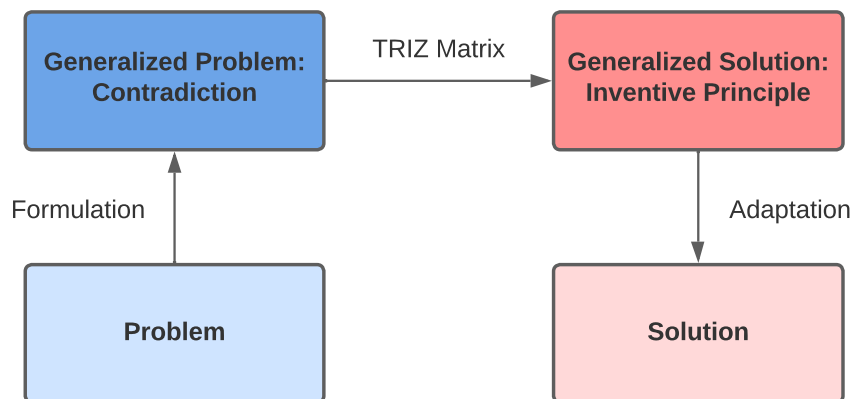


FIGURE 2.1: TRIZ stages for problem solving

2.2 Patent Mining

A patent is a legal document which mostly contains text. Patent mining is therefore strongly correlated with text mining. However, the structure of a patent and the way it is written (with legal terms and syntax) make it difficult to analyze automatically. Patent mining is, nevertheless, a common task in Natural Language Processing. This section highlights the different tasks linked to patents.

2.2.1 Patent structure

A patent aims at protecting an invention. It is a structured document because of its legal nature.

A number of items are associated with each patent such as the reference, inventors, filing date, date of issue. A description of the invention in its entirety is present with most often a brief presentation of the drawings, if any. The description informs about what is protected and what is free. The description generally contains a state of the art with previous solutions to the problem addressed. The claims define the legal scope of protection and delimit the monopoly of exploitation that the patent owner has. Not everything described in the patent is in the claims. The patent also includes a summary of the invention. This summary has no legal value. The summary must include the title of the invention, a concise summary of the essential

1. Weight of moving Object	21. Power
2. Weight of Stationary Object	22. Waste of Energy
3. Length of Moving Object	23. Waste of Substance
4. Length of Stationary Object	24. Loss of information
5. Area of Moving Object	25. Waste of time
6. Area of Stationary Object	26. Amount of substance
7. Volume of Moving Object	27. Reliability
8. Volume of Stationary Object	28. Accuracy of Measurement
9. Speed	29. Accuracy of Manufacturing
10. Force / Torque	30. Harmful factors acting on Obj.
11. Tension / Pressure	31. Harmful Side Effects
12. Shape	32. Manufacturability
13. Stability of Object	33. Convenience of Use
14. Strength	34. Reparability
15. Durability of Moving Object	35. Adaptability
16. Durability of Stationary Object	36. Complexity of Device
17. Temperature	37. Complexity of Control
18. Brightness	38. Level of Automation
19. Energy spent by of Moving Obj.	39. Productivity
20. Energy spent by of Stationary Obj.	

TABLE 2.3: TRIZ parameters

characteristics (it is therefore very close to the first claim) and possibly indications as to the use of the invention.

To define the field to which a patent belongs, the IPC (International Patent Classification) and CPC (Cooperative Patent Classification) have been introduced. The CPC is an extension of the IPC managed by the European Patent Office (EPO) and the US Patent Office (USPTO). It is a tree-based classification with 9 main sections, A-H and Y (Table 2.5), which are in turn subdivided into classes, subclasses, groups and subgroups. They allow to distinguish nearly 250,000 different sub-domains for CPC and nearly 100,000 for IPC.

The contents of the different parts (abstract, description, claims) are free and the form varies significantly depending on the patent writers. Moreover, the vocabulary is very technical and the legal nature of the patent is also felt in the unnatural formulations of the claims. Searching for information in patents therefore remains an important challenge despite its apparent organization.

2.2.2 Available datasets

The patent offices (especially EPO and USPTO) have understood the interest of providing open access data for research. This is why they provide data designed for easy exploitation.

For example, the USPTO has set up a platform allowing access to a multitude of datasets (<https://www.uspto.gov/ip-policy/economic-research/research-datasets>). This allows accessing to patent classification data, comments from patent officers on applications, for example, to find compilations of patents on particular subjects or access to an API allowing very broad access to all the data available to the USPTO.

In the same way, one can find datasets on the EPO site (<https://www.epo.org/searching-for-patents/data/bulk-data-sets.html>) for citations or full-text patents.

1. Segmentation	21. Skip
2. Taking Away	22. Turn the Harm to One's Good
3. Local Quality	23. Feedback
4. Asymmetry	24. Intermediary
5. Combining	25. Self-Service
6. Universality	26. Use of Copies (Copying)
7. Nested Doll	27. Cheap Short-Life Instead of Costly Long-Life
8. Anti-Weight	28. Mechanical Principle Replacement
9. Prior Counteraction	29. Pneumatic and Hydraulic Structures
10. Prior Action	30. Flexible Shells and Thin Films
11. Beforehand Cushioning	31. Porous Materials
12. Equipotentiality	32. Changing Color
13. Other Way Round	33. Homogeneity
14. Spheroidality	34. Rejecting and Regeneration of Parts
15. Dynamicity	35. Change of Physical and Chemical Parameters
16. Partial or Excessive Action	36. Phase Transitions
17. Another Dimension	37. Thermal Expansion
18. Mechanical Vibration	38. Strong Oxidizers (Strong Oxidants)
19. Periodic Action	39. Inert Atmosphere
20. Useful Action Continuity	40. Composites

TABLE 2.4: TRIZ inventive principles

- A. Human necessities
- B. Performing operations; transporting
- C. Chemistry; metallurgy
- D. Textiles; paper
- E. Fixed constructions
- F. Mechanical engineering; lighting; heating; weapons; blasting engines or pumps
- G. Physics
- H. Electricity
- Y. General tagging of new technological developments

TABLE 2.5: CPC sections

It is however complicated to find a dataset prepared by an office adapted for a non conventional use. The remaining solution is to download the weekly raw data containing the patents and to build a suitable dataset.

2.2.3 Patent search

The patent search includes many sub-tasks. Priority art search is the establishment of a state of the art on a technology, a field or a certain problematic. Patentability search refers to the search for any patent that could be seen as blocking the publication of a patent because it would protect elements of the patent. We can also mention the invalidity search, the infringement search or the legal status search. The key step of these searches is the formulation of a request corresponding to the need. For example, for a patentability search, only patents with a lower filling date are to be studied, whereas for prior art, all usable documents are exploited.

Most of the time, a simplification of the content can simplify the query. (Shinmori and Marukawa, 2003) propose for example to simplify the content of the claim by identifying groups of words linked by predefined relations. These relations are deduced from the morpho-syntactic structure of the sentences. Terms present in several sections of the patent can also be considered as more important (Xue and Croft, 2009) and can be used to create better queries. Finally, queries can be composed of sub-queries for each patent part (Konishi, 2005) or a single query applied to several parts (Mahdabi et al., 2012; Mahdabi et al., 2011).

2.2.4 Patent classification

Automatic patent classification facilitates the management of patents and also avoids time-consuming human analysis of patents by patent officers. It aims at associating each patent to an IPC or CPC domain.

Bag Of Words representations can be used to classify patents (Larkey, 1997). From the representations, patents are grouped by domain with clustering algorithms like kNN clustering or the category can be predicted from a classifier. Deep neural networks are also widely used as they can build very good representations of patents (Cho et al., 2014; Devlin et al., 2019; Peters et al., 2018). (Abdelgawad et al., 2019) compares several approaches including BERT, ELMo, a CNN and a hierarchical SVM and highlights the importance of the choice of hyperparameters for optimization. The Hierarchical SVM is built on the basis of the IPC classification model. Hierarchical classification is also employed in (Zhu et al., 2020; Shalaby et al., n.d.) with recurrent networks. (Risch, Garda, and Krestel, 2020) also focuses on hierarchical classification. However, the proposed approach is based on a Seq2Seq (Sequence to Sequence) model with an encoder and a decoder. The decoder iteratively generates the domains and subdomains of the patent classification. The architecture is based on CNNs, Transformers and LSTMs. BERT (Lee and Hsiang, 2020b) or Doc2Vec (Lu et al., 2020) are used for the classification and the similarity computation between employed technologies.

2.2.5 Generation tasks

Almost all works on patents are focused on patent landscaping (most often with unsupervised clustering techniques) or on domain classification with supervised techniques. We highlight the existence of a single work (Lee and Hsiang, 2020a) that deals with the automatic generation of claims in order to generate new ideas. It is however a task whose level of abstraction is relatively close to the generation of images from a statement (Gregor et al., 2015; Dosovitskiy, Tobias Springenberg, and Brox, 2015) which is more successful in other fields.

2.3 TRIZ and patent mining

Patents can be exploited to serve creative action by providing targeted information on existing inventions with text mining techniques. The mining of TRIZ parameters and inventive principles, in particular, have already been attempted.

2.3.1 Contradictions and TRIZ parameters extraction

In (Chang, Chang, and Wu, 2017) a method to retrieve TRIZ parameters is introduced. They assume that a patent solves one contradiction and two TRIZ parameters are therefore improved (or at least not degraded). The method consists in key phrases detection. One of the parameters is supposed to be in a sentence similar to "to be prevented from worsening" and the second parameter in a sentence with "to be improved". The detection method is therefore very limited and it works only for Chinese patents as the syntax does not vary much.

(Souili and Cavallucci, 2017) use linguistic tools to extract concepts related to Inventive Design ontology from patents, i.e. problems and partial solutions. Sentiment analysis is used, as a solution often lies in a "positive" sentence while a problem lies in a "negative sentence". The data extracted may then be represented in graphs (see Figure 2.2).

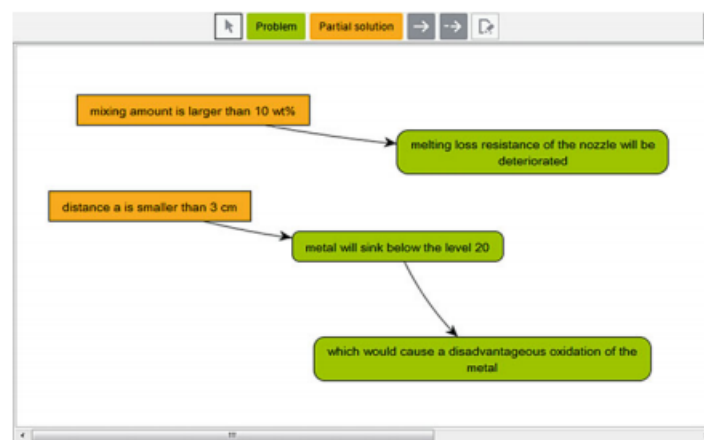


FIGURE 2.2: Problems and partial solutions represented in a graph

Following the work of (Souili and Cavallucci, 2017), (Berdyugina and Cavallucci, 2022a) proposed an approach based on antonyms and topic modeling to identify potentially conflicting parameters in patents from the problems and parameters isolated by (Souili and Cavallucci, 2017).

In (Yanhong Liang, Runhua Tan, and Jianhong Ma, 2008), a methodology to retrieve patents according to the contradictions they solve is presented. Nevertheless, the presented method uses Wordnet dictionary but it is unclear as to why, considering that no attempt to test this approach has been made. Therefore, it cannot be really considered as a prior art method. The same authors published another approach (Liang et al., 2009) one year later dealing with patent classification. This time, it was not a classification in accordance with the solved contradictions but in accordance with the inventive principle. Their methodology is described in Figure 2.3. First, the initial contradiction must be found. The most suitable inventive principles are then chosen and patents potentially related to the inventive principle are suggested to the user. To measure relevancy of patents, a traditional TF-IDF algorithm with Chi-square method are used.

In (Cascini and Russo, 2007), a framework to detect the solved contradictions is presented. The worsening factor is supposed to be located in the background of the invention / state of the art and the improving factor in the claims. Nevertheless, the method in use seems to be limited (keywords approach) and no numerical results were shown.

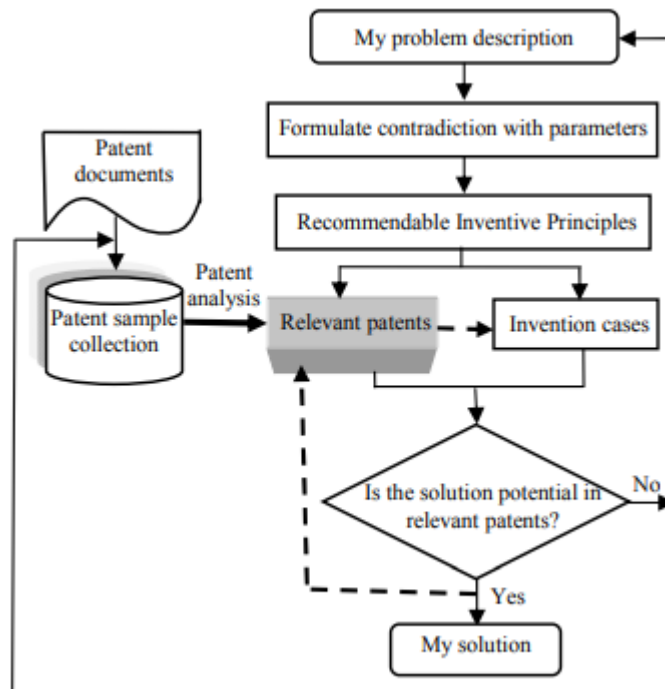


FIGURE 2.3: Methodology for problem solving according to (Liang et al., 2009)

2.3.2 Inventive principles, physical effects and solution mining

TF-IDF (Term Frequency-Inverse Document Frequency) and LDA clustering may be utilized for physical effects retrieval (Korobkin, Fomenkov, and Kravets, 2017).

(Loh, He, and Shen, 2006) show an attempt to classify patents in accordance with the inventive principle they use to solve a problem. A very small database is used (200 patents) and the patents are only classified using 6 inventive principles due to a lack of precision. (He and Loh, 2008) also try to classify patents as a function of the inventive principles. In this case, all the inventive principles are used but, in order to deal with the lack of data and to make the training possible, groups of inventive principles are created. The purpose of the method is to recognize which groups of inventive principles are linked to the patents. Even with this simplification, the results are noisy with a very low recall.

In (Liang and Tan, 2007), the authors are presenting another methodology, to achieve the same purpose, which is based on keywords. The idea is to recognize patterns associated with the problems or the way of solving these problems.

In (Wang et al., 2016), patent parameters of a particular process are mined using semantic databases. These parameters are then associated with the general parameters from TRIZ matrix. Contradiction solving principles are also clustered to build a new TRIZ matrix with the process patents (see Figure 2.4).

(Ni, Samet, and Cavallucci, 2022) propose a solving method based on problems matching. Their purpose is to avoid mining contradictions. Thus, from an initial problem, their model searches for semantically similar problems in patents. If problems are similar, the solution of the solved one should also apply to the non-solved one. It would at least give an insight on how to solve it. A question answering is coupled with the matching system to find the answer of the problem in the matched patents.

Weakening parameters Strengthening parameters	1.Work- piece structure	2.Process efficiency	3.Tooling complexity	4.Fixture clamping force	5.Fixturability of workpiece	6.Manufactu- ring quality	...
1.Workpiece structure	+	6, 11	1, 6, 7, 9, 12	1, 8, 10		9	
2.Process efficiency	3, 6	+	2, 8, 9	1	6, 8, 11	1, 12	
3.Tooling complexity	4, 5, 9	3, 8, 12	+	11, 12	3		
4.Fixture clamping force		1, 7	5, 10, 12	+	4, 7	3	
5.Fixturability of workpiece	2, 5			7, 12	+	1, 6, 13	
6.Manufactu- ring quality	12	3, 7	9	5		+	
.....							

FIGURE 2.4: Specialisation of TRIZ matrix (Wang et al., 2016)

TABLE 2.6: Patent Indicators and Explanations

Patent Indicator	Explanation
Number of Inventors (NI)	The number of inventors involved in the patent.
Cited-Forward Citations with no Family (CFCNF)	Forward Citations that are not family-to-family cites.
Cited-Forward Citations with Family (CFCF)	Forward Citations that are family-to-family cites.
Cited-Backward Citations with no Family (CBCNF)	Backward Citations that are not family-to-family cites.
Cited-Backward Citations with Family (CBCF)	Backward Citations that are family-to-family cites.
Family Size (FS)	The number of countries in which the same invention is patented.

2.3.3 Estimation of inventiveness level

The estimation of the level of inventiveness is one of the main challenge for patents analysis in TRIZ domain. Nevertheless, very few methods were proposed.

(Li et al., 2012) present a natural language processing and citation metric-based method. Machine learning is used to infer the degree of inventiveness after a training on a dataset of 75 rated patents. The machine learning model takes as inputs the backward citations, the originality, the Backward citation tag but also the knowledge transfer measure. They get more than 70% accuracy which is relatively impressive considering the size of their dataset. (Lanjouw and Schankerman, 2004) and (Creemers et al., 1999) discuss the ways to measure patent quality and the link between patent value and the number of citations. (Jugulum and Frey, 2007) could be seen as an attempt to redefine Altshüller's matrix by replacing the inventiveness in the patents with the notion of robustness. The more an invention is robust, the more reliable it is.

(Ni et al., 2021) propose a ranking system for patents using an inventiveness criterion based on the five indicators presented in Table 2.6. Nevertheless, it must be noted that this definition of inventiveness is not linked to the original definition of Altshuller using the distance between the problem's domain and the solution's domain.

Summary and conclusion

Table 2.7 summarizes all the contributions in TRIZ domain linked to text or patent mining.

The different approaches are organized following their purpose and their date. We highlight that, prior to this work, only keywords analysis (Cascini and Russo, 2007) was attempted to retrieve contradictions. It consists in retrieving the phrases positioned after "increase" or "decrease" for example. Nevertheless, the two parts of the contradiction are, thus, processed independently and no real contradiction relation is extracted.

Following this work, (Berdyugina and Cavallucci, 2022a) extended the keyword-based contradiction mining with the use of antonyms and switched to a deep learning-based approach.

Concerning TRIZ parameter mining which is also part of a full contradiction mining process, several works address this challenge but mostly using keywords and key phrases which is not very relevant for most of the patents. Automatic language processing and patent mining are not new fields and can therefore motivate the choice of an approach. Contradiction extraction is the identification of two contradictory parameters (in the TRIZ sense) in a textual content. An unsupervised approach does not seem to be feasible here since the modeling of the notion of TRIZ contradiction is difficult and no clustering algorithm would allow the identification of these two parameters in contradiction. A supervised approach is therefore chosen.

It appears that the contradictions are understood thanks to the sentences that surround the parameters. For example, in the US06938300 patent, the two following sentences are considered: *When the stroller 1 moves over a lawn or uneven road surfaces, it is necessary for the stroller wheels to have a large diameter so as to ensure the comfort of the baby. However, if each of the front wheel assemblies 11 has two large-diameter front wheels 13, the total volume and weight of the stroller 1 will increase significantly so that it is difficult to push the stroller 1.* The notion of contradiction is included not in the parameters but in the relationship between the two sentences. This finding implied the development of a two-step approach. First, the modeling of the contradiction relations at the sentence-level and then the extraction of the parameters contained in the sentences in contradiction.

For sentence-level analysis, the closest NLP task is automatic summarization. It consists in selecting the important information in a text with respect to its purpose. The idea is to shift the initial summarization task to a TRIZ summarization which aims at identifying the contradiction relations between sentences. The automatic summarization can be based on sentence classification or on a generative model (see Chapter 1).

The extraction of parameters is similar to a token classification and very close to a Named Entity Recognition task. An alternative way to avoid mining the parameters would be to use unsupervised learning. The ultimate purpose of this approach is to link a contradiction to out-of-domain patents. This link can be made without using the parameters but via clustering between the sentences of the contradictions. The idea would be to find a set of n clusters of "contradictory" sentences with for example K-means. Each cluster would correspond to a parameter contained in each sentence of the cluster. The advantage is that the original TRIZ parameters are not used to link different domain parameters. Indeed, there is no actual scientific proof showing that these original parameters are the best generic parameters to compare parameters from distant technical domains. The limit is that clustering seems to be difficult to achieve precisely because the sentences come from different domains.

	Purpose	Approach
(Souili and Cavallucci, 2017)	Problems / Partial Solutions / TRIZ Parameters	Keywords, Sentiment Analysis
(Wang et al., 2016)	Process Parameters	Keywords, WordNet
(Chang, Chang, and Wu, 2017)	TRIZ Parameters	Keywords
(Cascini and Russo, 2007)	Contradictory TRIZ Parameters	Keywords, Patterns
(Berdygina and Cavallucci, 2022a)	Contradictory TRIZ Parameters	Keywords, LDA
Berdygina and Cavallucci, 2022b	TRIZ Parameters	Cause-Effect, BERT
(Loh, He, and Shen, 2006)	Inventive Principles	KNN, SVM, Naive Bayes
(Liang and Tan, 2007)	Inventive principles	Keywords
(He and Loh, 2008)	Inventive Principles	SVM, Naive Bayes, Decision Tree
(Liang et al., 2009)	Inventive Principles	Bag of Words
(Korobkin, Fomenkov, and Kravets, 2017)	Physical effects	LDA, TF-IDF
(Cremers et al., 1999)	Patent value	Manual Analysis
(Lanjouw and Schankerman, 2004)	Patent value	Multiple Indicator Model (Linear)
(Jugulum and Frey, 2007)	Inventiveness / Robustness	Manual Analysis
(Li et al., 2012)	Inventiveness	ANN
(Ni et al., 2021)	Inventiveness	MCDA
(Ni, Samet, and Cavallucci, 2022)	Problem/Problem matching	LSTM

TABLE 2.7: Summary of the state of the art in TRIZ domain

	Noise sensitive (From best to worse)	Sub-tasks	Need for additional knowledge
Contradictory Parameters Mining	3	X	X
2 Steps approach with Clustering	2	✓	X
2 Steps approach with TRIZ Params. mining	1	✓	✓

TABLE 2.8: Summary of the possible approaches

To choose the best approach we compare different methods with the prism of how to link a targeted/input contradiction to processed patents in order to highlight possible solutions. The main approaches we identified are shown in Table 2.8 with qualitative evaluation on how the approach would be sensitive to noise, whether it implies sub-tasks and therefore potentially several datasets, and finally, whether it requires additional (uncertain) knowledge (here TRIZ original parameters). Noise is understood as "how the context may influence the results".

The first method consists in directly mining contradictory parameters from a textual content and matching them with the parameters of the input contradiction. A patent contains dozens of parameters. Mining two of them, considered as contradictory, seems highly difficult especially as contradiction information lies in the sentences around the parameters. This method is therefore labeled as the most sensitive to noise. To match the targeted parameters it would also require the use of the original TRIZ parameters (for cross-domain matching) or a clustering method which adds a layer of complexity.

The second method consists in a 2-steps approach. First contradiction relationships are identified using sentence-level information. Then, clustering is used to match each sentence/part of the contradiction to a parameter of the input contradiction. Clustering methods using sentences drawn from different patents from different domains would most likely perform poorly due to the words around the parameters. This can be mitigated with a syntactic approach by keeping only nouns, for instance, but parameters are often described with several words and not only nouns.

The third method consists in mining specific parameters from the sentences involved in the contradiction before translating them to TRIZ parameters and comparing them with the input parameters. We consider that supervised learning suffers less from this cross-domain analysis if the datasets are properly built (with multi-domain patents). That is why the fully supervised approach is ranked first for this criterion. Nevertheless, supervised training implies more implementation complexity and dataset creation for each supervised sub-task.

Considering all the criteria, we finally chose the two-step approach, fully supervised for sentence and word-level mining. As no existing contradiction mining approach was found at the time of this work, we compare our approach to classical summarization approaches. For parameter mining, the same process is applied and we compare our model to classical NER approaches as well as to the existing state of the art approaches.

Chapter 3

Extractive summarization for TRIZ contradiction mining

Contents

Introduction	35
3.1 Dataset	35
3.1.1 Parts of interest	35
3.1.2 Labeling details	36
3.2 Extractive Approach	37
3.2.1 Baseline approach: SummaTRIZ	37
3.2.2 PaGAN: Multi-level patent classification and semi-supervised contradiction mining	38
3.3 Experiments	43
3.3.1 Metrics	43
3.3.2 Hardware and experimental details	43
3.3.3 Results	47
3.3.4 Case study	49
Conclusion	50

Introduction

The state of the art has identified that summarization approaches can go well with contradiction mining. A first approach with extractive summarization is implemented. This first model is inspired by traditional summarization models and in particular the one of Liu and Lapata, 2019. Our contributions to improve the performance of this model thanks to a multi-level classification and a semi-supervised learning are, then, presented.

A TRIZ contradiction dataset has been created. It consists of 1600 US patents from which sentences containing the contradiction parameters are extracted. The construction and composition of the dataset are detailed in section 3.1.

3.1 Dataset

3.1.1 Parts of interest

Our attention is focused on *state of the art* part of patents. This part details what is at stake in the invention. The state of the art usually presents the purpose of the

invention and the difficulties encountered by the current prior art solutions. In order to mine the contradiction (between parameters) solved by the patent, several scenarios are possible. A starting problem can be presented as follows: *"When primary antioxidants, such as hindered phenols, are utilized, polymers can have a more yellow color than unstabilized polymers, therefore decreasing the commercial value of polymers"*. This sentence contains a parameter which worsens, in this case the *commercial value*. Solutions to this problem are then presented but they have drawbacks such as: *"Applications employing phosphite additives can result in a reduced quality in the physical properties of polyolefins"*. In this second sentence, a second parameter is listed which can be degraded by the application of the solution. Therefore, the solution mentioned in the state of the art leads to an improvement of the *"commercial value"* parameter but leads also to the degradation of the *"physical properties"* parameter. Consequently, there is a contradiction between these two parameters. In the same way, in the following example, a parameter in need of improvement is presented in the state of the art: *"This necessitates proper positioning of the upwardly-extending portion of the below-ground structure"*. This parameter is, therefore, the positioning. The drawbacks of the prior art solutions form a contradiction with this parameter as, for instance, the risks of injuries: *"Therefore, repetitively placing, removing and re-placing such device in the process of determining how best to complete the upper end of the upwardly-projecting portion of the below-ground structure involves considerable physical strain and accompanying risks"*. Finding this contradiction allows formulating the problem in an original way and an in-depth description of the object of the invention which, by hypothesis, responds to this (or these) contradiction(s). The sentences containing the parameters which need to be improved or the parameters involved in the initial problem are the **first part of the contradiction**. The drawbacks of the prior art solutions constitute the **second part of the contradiction**. This choice to separate the two parts of the contradiction is motivated by the will to identify the direction of the contradiction. Indeed, Altshuller's matrix presenting the inventive principles as a function of the contradictions between parameters is not symmetrical. Therefore, if a parameter *A* is in contradiction with a parameter *B*, the inventive principles that are statistically the most used, and therefore the types of solutions, are different than when *B* is in contradiction with *A*.

3.1.2 Labeling details

The state of the art parts of 1600 patents from the United States Patent Trademark Office (USPTO) was labeled to train the sentence classification model. The labeled patents come from all domains. As the patents do not always contain a proper state of the art and do not always contain a contradiction, these 1600 patents are sampled from an initial pool of about 15000 patents. Sentences can belong to three different classes: *First part of the contradiction*, *Second part of the contradiction* and *rejection class*. The sentences containing no evaluation parameters are assigned to the rejection class. The patents were labeled by a team of human experts. Details on the average number of sentences and labels in the dataset can be found in Table 3.1. We notice that the dataset is slightly unbalanced since there are 60% more examples for the second parts of contradictions than for the first parts.

Here is an example of an annotation with the patent US6938300B2:

This invention relates to a stroller, and more particularly to a wheel assembly for a stroller, which includes a single wheel.

...

Patents	Sent. – Sent./doc	1 st – 1 st /doc	2 nd – 2 nd /doc
1600	28732-17.96	2265-1.42	3714-2.32

TABLE 3.1: Details on the summarization dataset

In each of the front wheel assemblies 11, since the forward force A is located midway between two frictional forces B that are generated between the ground and the front wheels 13 and since the direction of the forward force A is parallel to those of the frictional forces B, the stroller 1 can advance along a straight path 16. When the stroller 1 moves over a lawn or uneven road surfaces, it is necessary for the stroller wheels to have a large diameter so as to ensure the comfort of the baby. However, if each of the front wheel assemblies 11 has two large-diameter front wheels 13, the total volume and weight of the stroller 1 will increase so significantly that it is difficult to push the stroller 1.

In this patent one can spot a first parameter in the blue sentence, comfort, which is improved by the use of large diameter wheels. Another parameter is then degraded, the ability to push the stroller, which belongs to the second part of the contradiction, in red.

Several sentences containing the same parameters of the contradiction may be found in a patent. In such cases, all these sentences are extracted. Thus, the conflicting parameters can possibly be extracted from several different pairs of sentences. Finally, if a sentence contains both parts of the contradiction, i.e. both parameters in the contradiction, it is classified as “First part of the contradiction” but also as “Second part of the contradiction”. Patents do not necessarily contain a contradiction, either because the writer does not give the necessary information or because the patent does not give a solution to a contradiction. It is therefore necessary to filter the patents through the prism of the presence or absence of a contradiction. This analysis at the document level is made possible by selecting 1600 patents that do not contain a contradiction in order to train a “filtering” model on the patents. The dataset containing both contradiction and no contradiction patents can be accessed through this [link](#).

3.2 Extractive Approach

3.2.1 Baseline approach: SummaTRIZ

The baseline model, called SummaTRIZ, is based on an extractive summarization model from Liu and Lapata, 2019. SummaTRIZ architecture is shown in Figure 3.1. BERT encoder is used in this model. BERT takes a series of tokens as input. Each sentence is separated by a special token [SEP]. Another special token [CLS] is used to represent each sentence. BERT also takes as input indicators of positions of tokens and sentences called positional embeddings and segment embeddings. The input embeddings then pass into a series of Transformer layers based on attention mechanisms that allow extracting the salient information. BERT builds a contextual representation for each input token. These representations integrate a maximum of information coming from adjacent tokens.

A Transformer layer on top of BERT, which only takes as input the token representations [CLS], which are thus the sentence representations, allows having global attention on the whole sequence even if this one was longer than the 512 tokens limit

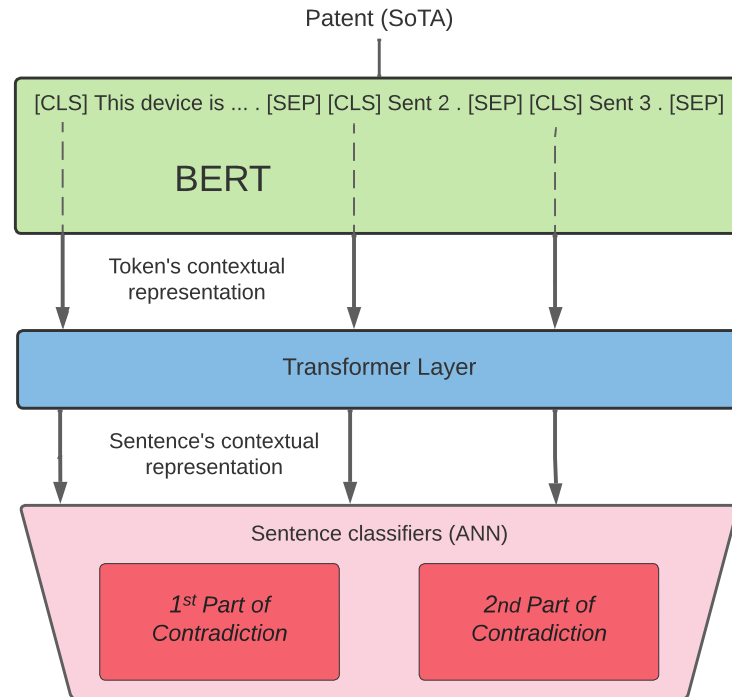


FIGURE 3.1: Baseline approach: SummaTRIZ (Guarino et al., 2020)

for BERT. The output of this last layer is thus a contextual representation of each sentence. The limit is set to 1500 tokens to fit the length of the patent's state of the art parts. This layer's purpose is also to model the relationships between sentences.

SummaTRIZ model is pre-trained on an extractive summarization task of press articles with the CNN/DailyMail dataset (Hermann et al., 2015). The objective is to train the attention layer above BERT so that it is able to build a representation containing the essential information in the input sequences. Indeed, the training of this layer requires a large variety of documents. Data are cleaned and tokenized using **Stanford-Core-NLP** tokenizer before the standard BERT's tokenizer. This process was used to pre-train the model on CNN/DailyMail dataset and it was therefore reused for patent processing so that the pre-trained model operates in the same conditions.

Two 2-class ANN classifiers are used, on top of the Transformer layer, to predict whether each sentence belongs to the first part of the contradiction, the second part of the contradiction or to none of them. Two different classifiers are used because there is a non-zero probability that a sentence contains the whole contradiction and thus should be classified as the first and second part of the contradiction at the same time.

3.2.2 PaGAN: Multi-level patent classification and semi-supervised contradiction mining

The limitation of the baseline model is that it does not predict whether a document contains a contradiction or not. Thanks to the experience gained from the labeling process of our dataset, we estimate that only 10% to 15% of the 7 million available patents contain a mineable contradiction. Therefore the addition of a document-level analysis would allow to automatically eliminate the patents that do not contain a contradiction.

Document-based approach for contradiction mining

We propose four different models for this part which cover the main tendencies in document classification: a probabilistic model, a model based on a recurrent network, a Transformer-based model, and an ANN-based model.

Probabilistic model In this model, we consider that the belonging of a sentence S_i to either the first part or the second part of the contradiction are independent events and thus:

$$P_c(S_1...S_n) = \max_{1 \leq i \leq n} (P_{c1}(S_i)) * \max_{1 \leq i \leq n} (P_{c2}(S_i)) \quad (3.1)$$

with $P_c(S_1...S_n)$ the probability that a contradiction is in a sentence sequence $S_1...S_n$. $P_{c1}(S_i)$ is the probability that sentence i is the first part of the contradiction and $P_{c2}(S_i)$ the probability that sentence i is the second part of the contradiction.

Recurrent model The probabilistic model is limited in performance because of its postulate, which is too strong. A more global analysis of the document seems therefore necessary. The main difficulty encountered is the variable length of the documents. One possibility is to use recurrent networks. A LSTM (Hochreiter and Schmidhuber, 1997a) or a GRU (Chung et al., 2014) are used for this model. These recurrent networks take as input a sequence of sentence representations, in this case, the representations of all sentences in the state of the art. The LSTMs contain a memory vector called cell state which allows them to select the "useful" information during the iterations. The cell state is modified at each inference (Equation 3.2), to take into account the previous state and the input.

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (3.2)$$

f_t is the forget gate which selects the information to keep in the cell state, i_t is the input gate that selects the information to add in the cell state given the input, and g_t is computed with the input and the previous output. In our case, the "useful" information are the features related to the presence or not of a contradiction. The features are learned by BERT encoder, so the recurrent network only has to extract the right ones. This is why only one unidirectional cell of LSTM is used. This allows to limit the number of parameters without affecting the results.

Transformer model The recurrent model has several drawbacks such as its run time or the loss of information due to successive iterations. A Transformer layer does not have these drawbacks. In this model, the Transformer layer takes the sequence of sentence representations as input and its first output is the document classification score.

ANN model The recurrent and Transformer models do not take into account the results related to the classification of sentences. We, therefore, present a last model based on a Multi-Layer Perceptron. The idea is to keep only decisive information in the decision-making concerning the existence of a contradiction. In this case, because the encoder allows building a very precise contextual representation of the sentences, only the representations associated with the two sentences forming the contradiction are necessary for the decision-making. A Multi-Layer Perceptron is

used for the classification of the document from the representations of the two sentences having the maximum probabilities of belonging to the contradiction (one sentence for the *first part* and one sentence for the *second part of the contradiction*). Thus, the probability $P_c(S_1 \dots S_n)$ that a contradiction is present in the document becomes:

$$P_c(S_1 \dots S_n) = ANN(\arg \max_{1 \leq i \leq n}(P_{c1}(S_i)), \arg \max_{1 \leq i \leq n}(P_{c2}(S_i))). \quad (3.3)$$

The model is less likely to overfit despite the small amount of data since, for the same documents, its inputs vary during the learning process. Indeed, as the sentences scores evolve during the training, for a same document, the selected sentences which go through the ANN model change. It causes the encoder to integrate the contradiction information in all "main" sentences. The ANN decision is, thus, easier.

A mass of several million of unlabeled patents is available. The possible use of unlabeled data led us to explore the potential of semi-supervised training to improve baseline results. We choose to implement a Generative Adversarial Network (GAN) to optimize the model's fine-tuning via semi-supervised learning and cope with the training dataset's size.

Semi-supervised contradiction mining

Semi-supervised learning aims to improve the generalization of a model using unlabeled data. It also improves the quality of the representations generated by the model (Weston, Ratle, and Collobert, 2008; Yang, Cohen, and Salakhutdinov, 2016; Kipf and Welling, 2017). Generative Adversarial Networks were introduced by Goodfellow et al. (Goodfellow et al., 2014). The purpose of GANs is to generate new data close to a target data distribution. The GANs can be adapted to semi-supervised learning (Salimans et al., 2016). Croce, Castellucci, and Basili, 2020 showed the efficiency of this method for classification.

A model called generator G generates dummy data and another model called discriminant D tries to distinguish the generated data among real but unlabeled examples.

The generator G learns to map input noise variables z to the real data distribution p_{data} . Its goal is, therefore, to minimize $\log(1 - D(G(z)))$. The discriminant, on the contrary, tries to maximize $\log(1 - D(G(z)))$ while associating the right labels to the real data x , i.e. maximizing $\log(D(x))$.

Thus, D and G play a two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \quad (3.4)$$

Model training An adversarial training for the sentence-level classifiers is implemented as described by Salimans et al. (Salimans et al., 2016). An additional class is created to introduce the probabilities of input data fitting to the target distribution p_{data} . Sentence classifiers have therefore three output neurons: the initial contradiction's classification neurons and another neuron which outputs the probability for the document to be fake (see Fig. 3.2). The "contradiction" neurons are therefore involved in the supervised loss (D_{sup_1} or D_{sup_2}) for the two contradiction's classification classes C and \bar{C} (the sentence belongs to the part i of the contradiction or not, Equation 3.5) while the "adversarial" neuron is involved in the unsupervised losses for the *fake* F and *real* \bar{F} classes (Equations 3.6 and 3.7).

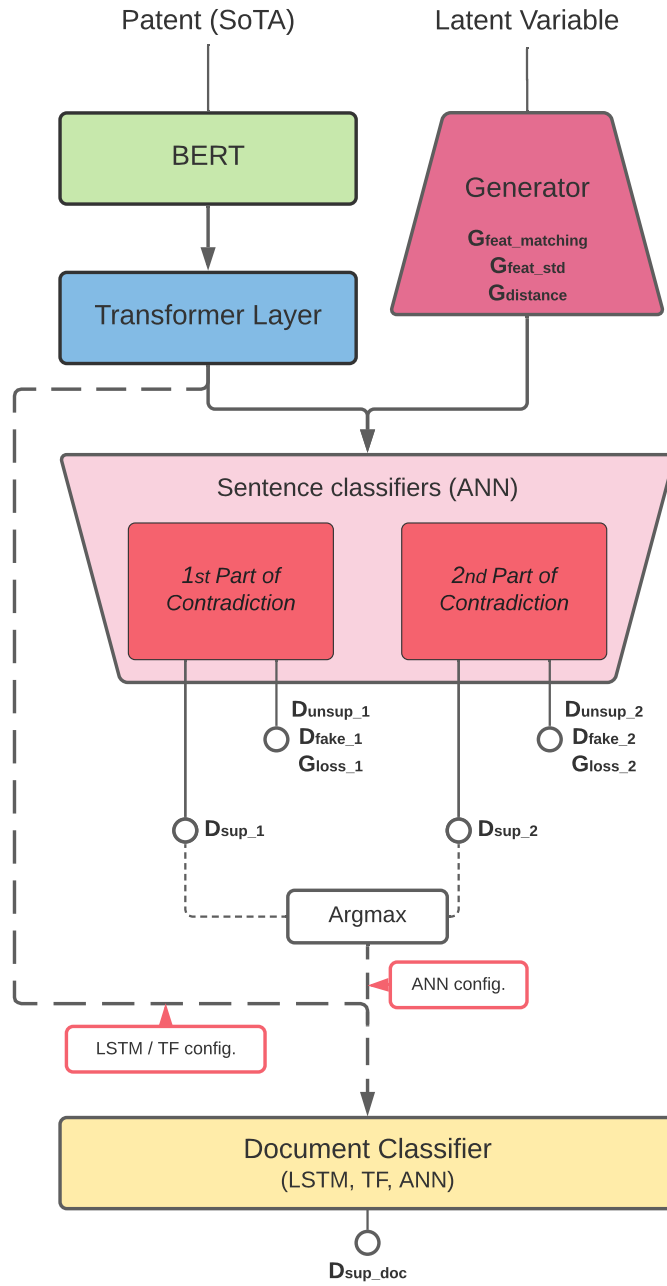


FIGURE 3.2: PaGAN architecture

$$D_{sup_i} = E_{x,y \sim p_{data}} [-\log(P(\hat{y}_{s_i} = y_{s_i} | x, y_{s_i} \in (C, \bar{C})))] \quad (3.5)$$

$$D_{unsup_i} = E_{x \sim p_{data}} [-\log(P(\hat{y}_{s_i} = y_{s_i} | x, y_{s_i} = \bar{F}))] \quad (3.6)$$

$$D_{fake_i} = E_{z \sim p_z} [-\log(P(\hat{y}_{s_i} = y_{s_i} | x, y_{s_i} = F))] \quad (3.7)$$

D_{unsup} relates to how good the model is at classifying the real data as real data. D_{fake} relates to how good the model is at classifying the fake data as fake data. The

sentences used for the unsupervised losses come from unlabeled patents. As the losses are computed at a sentence level, not many documents are needed (we only use 1/10 of the number of labeled documents). D_{unsup} is therefore back-propagated through the Transformer Layer and BERT encoder. This implies that the encoder learns to integrate new information in the sentence representations to allow better recognition of the "real" data. This richer representation of patents' sentences induces an easier classification for the contradiction which is our main purpose. This mechanism also allows minimizing overfitting. This is even amplified by the fact that two non-supervised losses are computed and back-propagated at the same time (one for each classifier).

The document classifier has only one output neuron to decide between documents that contain a contradiction (Dc class) and those that do not (\overline{Dc} class). It is thus involved in a single supervised loss:

$$D_{sup_doc} = E_{x,y \sim p_{data}} [-\log(P(\hat{y}_d = y_d | x, y_d \in (Dc, \overline{Dc})))] \quad (3.8)$$

As the number of labeled documents is lower than the number of labeled sentences, the document classifier tends to learn faster than the sentence classifiers. An experimental coefficient of 0.1 is therefore applied to D_{sup_doc} so that the learning curves of all classifiers can match.

Generator architecture and training A few different architectures are implemented for the generator: a fully connected network, a LSTM, and a Transformer. The comparison of these different architectures is shown in section 3.3.

The generator, in our case, generates sequences of sentence representations that are plausible in order to deceive the sentence-level classifiers but, as, BERT uses the context to encode the sentences, the generator must also generate a plausible context. The loss associated with the generator combines both losses from sentence-classifiers (G_loss_1 and G_loss_2 in Fig. 3.2):

$$G_{loss} = \sum_{i=1}^2 E_{z \sim p_z} [-\log(1 - P(\hat{y}_{s_i} = y_{s_i} | x, y_{s_i} = F))] \quad (3.9)$$

To ensure that the generator converges to the right distributions, we add two feature matching losses. G_{feat_mean} (Equation 3.10) ensures that the generated representations is close to the training data and G_{feat_std} (Equation 3.11) also ensures that the generated documents are not a single representation of a sentence repeated n times. We, thus, add a constraint on the variety of representations in the same document.

$$G_{feat_mean} = \left\| E_{x \sim p_{data}(x)} f(x) - E_{z \sim p_z(z)} f(G(z)) \right\| \quad (3.10)$$

$$G_{feat_std} = \left\| \sigma_{x \sim p_{data}(x)} f(x) - \sigma_{z \sim p_z(z)} f(G(z)) \right\| \quad (3.11)$$

Finally, to bring variety in the generated documents, we introduce a minimization of the similarity between generated documents via a cosine similarity measure:

$$Cos_{similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|} \forall A, B \in \mathbb{R}^n \quad (3.12)$$

At each iteration, a constant number of documents is generated but they include a random number X of representations with $X \sim \mathcal{N}(\mu, \sigma^2)$ and $\mu = 8, \sigma = 2$. All the documents generated at an iteration i have nevertheless the same length X_i . The

computation of the similarity between documents can be done relatively easily by using the similarity between sentences. The similarity between two documents is, thus, defined as the sum of the similarities between the sentences of these documents.

In practice, this additional strong constraint, since it forces the representations to be very different, does not interfere with the convergence of the generator while avoiding mode collapse.

3.3 Experiments

3.3.1 Metrics

Contradictions mining relates to a classification problem. This is why the classical classification metrics (Accuracy, Precision, Recall, F1 score) are used for the evaluation. However, they are not sufficient to fully evaluate the mining performance. New metrics are therefore introduced and among them, a few are inspired from extractive summary metrics.

The metrics S and S_m are used to evaluate sentence-level analysis. A labeled document includes n_1 sentences for the first part of the contradiction and n_2 sentences for the second part of the contradiction. The metric S for a labeled document D is defined as the number of correct sentences for the first/second part of the contradiction in the n_1/n_2 sentences with the maximum probabilities of belonging to the *first/second part of contradiction*. S_m evaluates the number of correctly extracted sentences in the $n_1 + margin$ or $n_2 + margin$ best sentences depending on the part of the studied contradiction (we take $margin = 2$ for the experiments). These metrics allow us to assess the importance of the sentences that are extracted.

We call $E_1 = \{S_{10}...S_{1n_1}\}$ the set of the n_1 sentences labeled as *first part of contradiction* and $E_2 = \{S_{20}...S_{2n_2}\}$ the set of the n_2 sentences labeled as *second part of contradiction*. Each pair $S_{1i}S_{2j}$ forms a contradiction. We consider that a contradiction is extracted if:

$$\arg \max_{1 \leq i \leq n} (P_{c1}(S_i)) \in E_1 \quad (3.13)$$

$$\arg \max_{1 \leq i \leq n} (P_{c2}(S_i)) \in E_2 \quad (3.14)$$

$$P_c(S_1...S_n) > P_{threshold} \quad (3.15)$$

with $P_{c1}(S_i)$ the probability that sentence i is the first part of the contradiction, $P_{c2}(S_i)$ the probability that sentence i is the second part of the contradiction and $P_c(S_1...S_n)$ the probability that there is a contradiction to mine. The CO_{Found} metric evaluates the first two conditions so that it only takes into account the analysis of the sentences. The CO_{Valid} metric is used to evaluate the number of extractions that verify all these conditions, i.e. correct sentences are chosen and the document has a high probability of containing a contradiction. We take $P_{threshold} = 0.5$ for the experiments.

3.3.2 Hardware and experimental details

All the experiments are performed using a four RTX 2080Ti and Intel Core i9-9820X 3.30 GHz machine. Pytorch framework is used for all experiments.

Model	Loss	TP	FP	TN	FN	Accuracy	Precision	Recall	F1 score	S	S_m
SummaTRIZ _D	0.140	0	0	61959	2276	0.96	0	0	0	548	1158
SummaTRIZ _{TL}	0.115	576	510	61449	1700	0.97	0.53	0.25	0.34	1119	1711
Baseline _{ANN_D}	0.140	0	1	65248	2276	0.97	0	0	0	535	1149
Baseline _{ANN_{TL}}	0.115	575	482	64767	1701	0.97	0.54	0.25	0.35	1098	1710
PaGAN _{PROB}	0.112	575	457	71584	1701	0.97	0.56	0.25	0.35	1168	1736
PaGAN _{ANN}	0.112	532	414	71689	1744	0.97	0.56	0.23	0.33	1187	1760
PaGAN _{LSTM}	0.112	509	368	71704	1767	0.97	0.58	0.22	0.32	1186	1752
PaGAN _{TF}	0.113	649	592	71438	1627	0.97	0.52	0.29	0.37	1143	1759

TABLE 3.2: Sentence classification 1 (with LSTM generator)

Model	Loss	TP	FP	TN	FN	Accuracy	Precision	Recall	F1 score	S	S_m
SummaTRIZ _D	0.171	0	0	60526	3709	0.94	0	0	0	1750	2692
SummaTRIZ _{TL}	0.129	1814	815	59711	1895	0.96	0.69	0.49	0.57	2493	3127
Baseline _{ANN_D}	0.170	0	0	63816	3709	0.95	0	0	0	1766	2692
Baseline _{ANN_{TL}}	0.129	1849	881	62935	1860	0.96	0.68	0.50	0.57	2500	3131
PaGAN _{PROB}	0.120	2091	936	69672	1618	0.97	0.69	0.56	0.62	2619	3226
PaGAN _{ANN}	0.120	2288	1119	69551	1421	0.97	0.67	0.62	0.64	2626	3206
PaGAN _{LSTM}	0.118	2323	1156	69483	1386	0.97	0.67	0.63	0.65	2645	3213
PaGAN _{TF}	0.121	2327	1228	69369	1382	0.96	0.65	0.63	0.64	2631	3220

TABLE 3.3: Sentence classification 2 (with LSTM generator)

Model	Loss	TP	FP	TN	FN	Acc.	Pre.	Recall	F1 score	CO _{Found}	CO _{valid}
SummaTRIZ _D	-	0	0	1600	1600	0.50	0	0	0	153	0
SummaTRIZ _{TL}	-	386	135	1465	1214	0.58	0.74	0.24	0.36	582	213
Baseline _{ANN_b}	0.529	1274	490	1110	326	0.74	0.72	0.80	0.76	146	96
Baseline _{ANN_{TL}}	0.502	1275	438	1162	325	0.76	0.74	0.80	0.77	580	467
PaGAN _{PROB}	-	335	126	1474	1265	0.57	0.73	0.21	0.33	668	192
PaGAN _{ANN}	0.466	1335	431	1169	265	0.78	0.76	0.83	0.79	666	576
PaGAN _{LSTM}	0.481	1370	507	1093	230	0.77	0.73	0.86	0.79	654	567
PaGAN _{TF}	0.467	1345	427	1173	255	0.79	0.76	0.84	0.80	648	552

TABLE 3.4: Document classification (with LSTM generator)

Setup Generator	Loss	TP	FP	TN	FN	Accuracy	Precision	Recall	F1 score	S	S _m
LSTM _S	0.112	532	414	71689	1744	0.97	0.56	0.23	0.33	1187	1760
FC _S	0.113	575	493	71540	1701	0.97	0.54	0.25	0.34	1151	1760
TF _S	0.116	313	199	71873	1963	0.97	0.61	0.14	0.22	1157	1741
LSTM _{ALL}	0.114	561	438	71672	1715	0.97	0.56	0.25	0.34	1146	1729
TF _{ALL}	0.118	286	170	71863	1990	0.97	0.63	0.13	0.21	1132	1714
LSTM _D	0.112	462	326	71771	1814	0.97	0.59	0.20	0.30	1168	1736
TF _D	0.111	537	387	71730	1739	0.97	0.58	0.24	0.34	1182	1759

TABLE 3.5: Sentence classification 1 (with ANN document classifier)

Setup Generator	Loss	TP	FP	TN	FN	Accuracy	Precision	Recall	F1 score	S	S_m
$LSTM_S$	0.120	2288	1119	69551	1421	0.97	0.67	0.62	0.64	2626	3206
FC_S	0.118	2285	1135	69465	1424	0.97	0.67	0.62	0.64	2630	3231
TF_S	0.130	1735	690	69949	1974	0.96	0.72	0.47	0.57	2602	3192
$LSTM_{ALL}$	0.121	2285	1153	69524	1424	0.97	0.66	0.62	0.64	2639	3217
TF_{ALL}	0.129	1593	587	70013	2116	0.96	0.73	0.43	0.54	2598	3185
$LSTM_D$	0.118	1960	773	69891	1749	0.97	0.72	0.53	0.61	2663	3245
TF_D	0.118	2269	1060	69624	1440	0.97	0.68	0.61	0.64	2637	3231

TABLE 3.6: Sentence classification 2 (with ANN document classifier)

Setup Generator	Loss	TP	FP	TN	FN	Acc.	Pre.	Recall	F1 score	CO_{Found}	CO_{valid}
$LSTM_S$	0.466	1335	431	1169	265	0.78	0.76	0.83	0.79	666	576
FC_S	0.459	1363	466	1134	237	0.78	0.75	0.85	0.79	632	548
TF_S	0.508	1363	532	1068	237	0.76	0.72	0.85	0.78	630	547
$LSTM_{ALL}$	0.460	1384	467	1133	216	0.79	0.75	0.86	0.80	620	531
TF_{ALL}	0.512	1449	618	982	151	0.76	0.70	0.91	0.79	598	552
$LSTM_D$	0.471	1392	498	1102	208	0.78	0.74	0.87	0.80	641	575
TF_D	0.479	1418	592	1008	182	0.76	0.71	0.89	0.79	653	574

TABLE 3.7: Document classification (with ANN document classifier)

4-fold cross-validation is performed for all experiments (at the best of three trainings). All results, except losses, accuracy, precision, recall, and F1-score are summed up over the 4 folds. Losses are averaged out and the accuracy, precision, recall and F1-score are computed from the summed TP, FP, TN, and FN.

3.3.3 Results

The results of sentences' classification are shown in Tables 3.2 and 3.3. Results of document classification are sketched in Table 3.4.

SummaTRIZ (Guarino et al., 2020) is the only existing approach that has tackled contradiction mining. This model only allows sentence classification. In order to compare it to other approaches, we introduce the classification of documents in SummaTRIZ model with a probabilistic model (Part 3.2.2). SummaTRIZ_D model is trained on our dataset only. *TL* subscript refers to Transfer Learning, it indicates that the model is first trained on CNN/DailyMail dataset (extractive summarization task) and finetuned with our dataset. For a fairer comparison of the models and to highlight the contribution of the GAN, a SummaTRIZ model with an ANN document classifier (Part 3.2.2) is also introduced as *Baseline*. The influence of the document classifier (PROB described in 3.2.2, ANN described in 3.2.2, LSTM described in 3.2.2 and TF described in 3.2.2) is also studied with PaGAN. Note that for the sentence classifiers, we use an ANN with a single hidden layer.

At first, we notice that the dataset alone is not sufficient to achieve a correct level of performance. Indeed, very few sentences have a probability of belonging to *First part of contradiction* or *Second part of contradiction* greater than 0.5 which automatically brings most of the classification metrics to 0. Moreover, the ranking of the sentences made by SummaTRIZ_D and Baseline_{ANN_D} is much worse than for the other tested setups (other architectures and/or training mode). S and S_m are, in fact, almost twice as low as for the other candidates. This means that in the theoretical summaries provided by these models, very few sentences actually contain information about one or more contradictions. The results in terms of document classification are also lower than those of the other setups.

PaGAN shows slightly better sentence classification results for the traditional metrics but also for the new metrics S and S_m with an improvement of respectively 8% and 6% for S and 3% for S_m comparatively to the baseline using transfer learning (Baseline_{ANN_{TL}}). In terms of document classification and the number of contradictions extracted, the difference is much more visible. The loss is about 7% lower for the best configuration of PaGAN. For the contradictions found CO_{found} , in the sense of the good sentences selected regardless of the classification of the document, we see a maximum increase of 15.7% (from 580/1600 to 668/1600). Since document classification is better with GAN, this gap grows even larger when the whole model is considered with both sentence and document classification. Indeed, we observe an increase in the number of contradictions extracted and validated by the document-level classifier of more than 23% to go from 467/1600 to 576/1600 for the best setup of PaGAN. The contribution of adversarial training is thus clearly visible, both in terms of losses and metrics, and whether it is at the level of sentence or document classifications. It is also clear that, even if the GAN works only at the sentence level, better representations are learned by the encoder and it has a very positive impact on document classification.

Four different setups of PaGAN were tested with different document classifiers (Probabilistic model, ANN model, LSTM model, Transformer model). The naive probabilistic approach PaGAN_{PROB} is, unsurprisingly, very poorly performing for

document classification since even if an equivalent number of contradictions are found by sentence classification (CO_{found}), very few are actually validated by document classification (less than one third). The other three setups show very similar results. The most important metric is precision since the goal is to limit false positives as much as possible. The approach with the ANN, which presents at the same time high precision and a number of extracted contradictions slightly higher than the others, seems to be the best choice.

Finally, we studied the impact of the generator architecture as well as the training mode (see Tables 3.5, 3.6, 3.7). Several generator architectures have been implemented: LSTM, Transformer, Fully connected. A study of the impact of adversarial training has also been performed. Initially, only the sentence classifiers are involved in the adversarial training (index S added to the name of the generator architectures in tables 3.5, 3.6, 3.7). In another configuration, we integrate the document classifier to the discriminant which means that we add a term to the unsupervised losses for the discriminant and the generator (index ALL added to the name of the generator architectures in Tables 3.5, 3.6, 3.7):

$$D_{unsup} = E_{x \sim p_{data}} [-\log(P(\hat{y}_{doc} = y_{doc} | x, y_{doc} = \bar{F}))] + \sum_{i=1}^2 E_{x \sim p_{data}} [-\log(P(\hat{y}_{s_i} = y_{s_i} | x, y_{s_i} = \bar{F}))] \quad (3.16)$$

$$D_{fake} = E_{z \sim p_z} [-\log(P(\hat{y}_{doc} = y_{doc} | x, y_{doc} = F))] + \sum_{i=1}^2 E_{z \sim p_z} [-\log(P(\hat{y}_{s_i} = y_{s_i} | x, y_{s_i} = F))] \quad (3.17)$$

$$G_{loss} = E_{z \sim p_z} [-\log(1 - P(\hat{y}_{doc} = y_{doc} | x, y_{doc} = F))] + \sum_{i=1}^2 E_{z \sim p_z} [-\log(1 - P(\hat{y}_{s_i} = y_{s_i} | x, y_{s_i} = F))] \quad (3.18)$$

In a final setup, we remove the terms linked to the sentence classifiers so that only the document classifier plays the role of discriminant (index D added to the name of the generator architectures in tables 3.5, 3.6, 3.7). The Fully Connected FC generator is only a sentence generator. It takes as input a latent vector and generates a representation. Therefore, only the results of the first training configuration (adversarial training on sentence classifiers) are shown. The LSTM and the Transformer generators generate entire documents with linked sentence representations. That is why document-level adversarial training is relevant.

Tables 3.5 and 3.6 show that the architecture of the generator has an effect on the sentence classification metrics. The Transformer-based generators have high precision. Indeed, for the *first part of contradiction*, Transformers reach a precision of 0.61 on average against 0.57 for LSTM-based generators and 0.54 for the FC generator. For the *second part of contradiction* the average precision of Transformers is 0.71 against 0.68 and 0.67 for the LSTMs and FC generators). LSTMs show higher recall (0.23 for the *first part of contradiction* and 0.59 for the *second part of contradiction* versus 0.17 and 0.50 for the Transformers). The fully connected generator shows, as for LSTMs, a low precision but a high recall. Based on the sentence classification, the Transformer seems to be the best architecture since it limits the maximum number of false positives.

However, the observation is different for document classification (Table 3.7). Indeed, the LSTMs based and FC architectures show the best precision (0.75 on average

against 0.71 for the Transformers generators) while the Transformers have higher recalls. The document classification is important because it validates the sentence selection. Therefore, the best choice of architecture, validated by the metrics associated with the contradiction extraction CO_{found} and CO_{valid} is an LSTM.

The influence of the training mode is also visible since sentence-level training (*S*) goes for better recall for sentence classification and better precision for document classification and vice versa for document-level training (*D*). The dual-level training *ALL* is a compromise between both training mods but it does not improve the precision of document classification compared to the *S* training. Therefore, the latter appears to be the best training mode.

The best configuration for the generator is therefore an LSTM with sentence-level adversarial training to achieve maximum precision for document classification as well as a large number of good sentence-level extractions and contradictions mining.

3.3.4 Case study

We applied our model on patents related to hydrogen storage to automatically extract the main research issues/themes in this field. Here are a few results among the patents with the highest contradiction probabilities:

Positive examples

Patent US10106875 B2:

First part of contradiction: "While the maximum **filling pressure** of a vehicle-mounted hydrogen tank is currently 35 MPa, it is desired to increase the maximum filling pressure to 70 MPa in order to increase the **driving ranges** of fuel-cell vehicles to a level comparable to the driving ranges of gasoline vehicles."

Second part of contradiction: "However, a hydrogen pressure exceeding about 15 MPa increases the risk of hydrogen **embrittlement fracture** that may occur during service."

Patent FR2928991 A1:

First part of contradiction : "In order to allow the diffusion of hydrogen as a fuel for the transport or for the supply of the isolated sites, it is necessary to clear and safe storage systems and **weight and volume densities** sufficient."

Second part of contradiction: "However this significant gain in **volume density** of 68 % h₂ is not yet really exploited due to the increase of the **weight** of the structure serving as a reservoir."

Patent US10619794 B2:

First part of contradiction: "This configuration enhances the **rigidity** of the liner, so the high-pressure hydrogen tank can hold high-pressure hydrogen inside."

Second part of contradiction: "However, because the high-pressure hydrogen tank disclosed in jp-a -94 is a large tank shaped like a barrel, there are cases where the **cabin space and/or luggage space is reduced** to install the high-pressure hydrogen tank in a vehicle."

Negative examples

Patent US10024498

First part of contradiction: "Consequently, the use of the hydrogen energy has many benefits such as high **environmental protection** and **low pollution**."

Second part of contradiction: "Since the process of sequentially placing the aluminum boxes into the canister body and thermally-treating processes are **troublesome, time-consuming, labor-intensive, costly** and **power-consuming**, the **fabricating cost** of the hydrogen storage canister can not be reduced."

Patent US10511039

First part of contradiction: "However, since the vessel structure is changed, a material type and size of fins and tubes, and a loading method of a hydrogen storage material are **required to be changed**."

Second part of contradiction: "However, these methods may cause an increase in a **system volume** and an **energy loss**."

Patent US07108757

First part of contradiction: "While the world 's oil reserves are being rapidly depleted, the supply of hydrogen remains virtually unlimited."

Second part of contradiction: "Additionally, transfer of a large sized vessel is very difficult."

One can clearly see the presence of opposing evaluation parameters like the driving range and the risk of embrittlement fracture in US10106875 B2, or the rigidity and the required space in US10619794 B2. In each case, the improvement of the first parameter leads to the degradation of the second parameter. Sometimes, both parameters can be present in the same sentence (FR2928991 A1).

However, we note that the model can easily be misled by keywords such as "however" or "difficult" (Patents US10511039 and US07108757) and that it sometimes confuses the "high level" function of the object with the problems encountered to implement this function. For example in the patent US10024498 the model puts forward the protection of the environment with the use of hydrogen but not the challenges encountered to use hydrogen.

Conclusion

Extracting contradictory sentences in the context of TRIZ is a hard challenge due to the technical vocabulary and exogenous knowledge that are often required by experts to identify a contradiction. The contradictory parameters are, indeed, often not explicitly mentioned but only described. This makes the ML models more prone to errors when mining sentences with long descriptions of problems without real parameters.

Automatic summarization provides a first approach that shows encouraging results (1/3 of contradictions extracted). A first model directly inspired by the state of the art in automatic summarization has been implemented. We highlight that supervised learning is very quickly limited by the size of the constructed dataset. Nevertheless, we show that semi-supervised learning with a generative and multi-scale adversarial network can substantially improve the mining results.

To fully define a contradiction in the sense of TRIZ, the evaluation parameters (EPs) are missing. Only sentences can be labeled with this extractive summarization model. The object of the next chapter is therefore TRIZ parameter mining.

Chapter 4

Syntactic Conditional Random Field for TRIZ Parameter mining

Contents

Introduction	51
4.1 Conditional Random Fields and TRIZ parameter mining	52
4.2 Matrix-based Syntactic CRF	54
4.2.1 SynCRF-matrix	55
4.2.2 SynCRF-point	55
4.3 Potential-based Syntactic CRF	57
4.3.1 SynCRF-pos: Part Of Speech-based Syntactic CRF	57
4.3.2 SynCRF-context: context-based Syntactic CRF	58
4.4 Dataset and training	59
4.5 Experiments and results	60
4.5.1 Metrics and baseline approaches	60
4.5.2 Results	61
4.5.3 Discussion	65
Conclusion	66

Introduction

A contradiction is a particular relationship between two parameters. If sentences describing such relationships are mined, these parameters still need to be extracted.

The parameters consist of one or more words. The extraction of parameters is, therefore, similar in form to a Named-Entity Recognition task. The labels are, thus, adapted in accordance with the BIO policy with B (Begin) for the parameter start token, I (Interior) for parameters consisting of a single token or for tokens belonging to parameters and located after the start token, and O (Out) for tokens not belonging to a parameter.

Paper quality is improved by increasing cell##ulose pro##portion
 B-PE I-PE O O O O B-PA I-PA I-PA I-PA

Classical neural encoders provide a contextual representation for each token. A softmax classifier can be added with five output neurons (B and I for the action parameters and for the evaluation parameters, and O for the other tokens, i.e. B-PE, I-PE, B-PA, I-PA, O) to predict the label of each token.

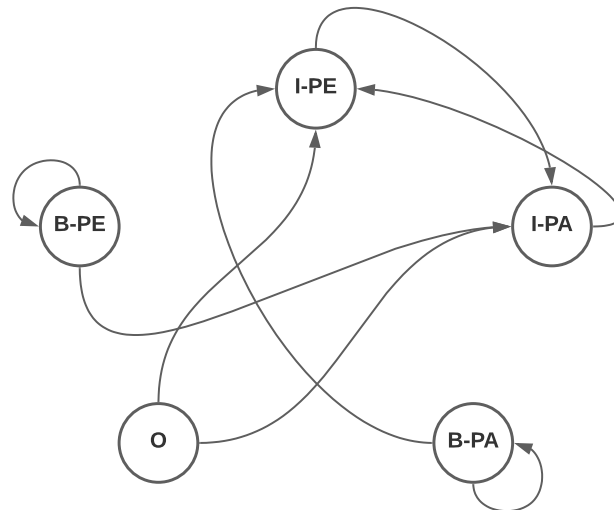


FIGURE 4.1: Impossible label transitions

In the case of TRIZ parameters, we notice that the syntactic structures introducing them are often similar (without being identical). This is mainly due to the evolution verbs (increase, decrease, improve...) which are often found near the parameters. An encoder with a softmax classifier will predict a label for each token without taking into account the labels of the neighboring tokens. However, in the previous example, we can see that if *Paper* is part of an *EP*, *quality* is likely to be part of it too. It therefore seems legit to try to introduce a dependency between the predicted labels with a Conditional Random Field (CRF).

In this chapter we will focus on TRIZ parameter mining with a CRF-based approach. A new CRF, called SynCRF, is introduced to take into account the syntactic specificity of patent sentences and achieve state of the art results in parameter mining.

Several transitions between labels are impossible (Figure 4.1), such as EP-B/EP-B or EP-B/AP-B. Indeed, since the action and evaluation parameters are nominal groups, they cannot be placed consecutively; they have to contain one verb in between. These dependencies are not easily modeled with a linear classifier but it is quite straightforward with a CRF.

4.1 Conditional Random Fields and TRIZ parameter mining

A Conditional Random Field (CRF) (Lafferty, McCallum, and Pereira, 2001) models the dependencies between neighboring variables (Chu et al., 2016). In classification tasks, the CRF model computes the conditional probabilities $P(Y|X)$ with Y the labels and X the observations. Each label depends on the current observation as well as on the preceding and the following labels (Markov property).

Let us assume that Y and X correspond respectively to a sequence of l labels and their corresponding sequence of l observations. $P(Y|X)$ is computed from each label

and observation of the sequence (considering that the labels are predicted independently of one another at first) with the following formula:

$$\begin{aligned}
 P(Y|X) &= \prod_{k=0}^{l-1} P(Y_k|X_k) \\
 &= \prod_{k=0}^{l-1} \frac{\exp(U(X_k, Y_k))}{Z(X_k)} \\
 &= \frac{\exp(\sum_{k=0}^{l-1} U(X_k, Y_k))}{Z(X)}
 \end{aligned} \tag{4.1}$$

with $Z(X)$, the partition function, i.e. the normalization factor computed from the sum of all possible numerators (for each possible labels sequence) and $U(X_k, Y_k)$ the *unary potential* referring to the likelihood that label Y_k is assigned given an observation X_k . $P(Y_k|X_k)$ is modeled with a normalized exponential as in a classical softmax output of a neural network.

If the dependency between two successive labels k^{th} and $k + 1^{th}$ is established, then a linking term could be added to $P(Y|X)$ and therefore could be written as follows:

$$\begin{aligned}
 P(Y|X) &= \prod_{k=0}^{l-1} \frac{\exp(U(X_k, Y_k)) \exp(T(Y_{k+1}, Y_k))}{Z(X_k)} \\
 &= \frac{\exp(\sum_{k=0}^{l-1} U(X_k, Y_k) + \sum_{k=0}^{l-2} T(Y_{k+1}, Y_k))}{Z(X)}
 \end{aligned} \tag{4.2}$$

with $T(Y_{k-1}, Y_k)$ the transition potential between label Y_{k-1} and label Y_k which is called the *pairwise potential*. The pairwise potential $T(Y_{k-1}, Y_k)$ refers to the likelihood of Y_k label being followed by Y_{k+1} . Pairwise potentials are usually stored in a matrix called *transition matrix*. When the CRF is associated to a neural encoder (i.e neural random fields (Peng, Bo, and Xu, 2009; Saha, Saha, and Bhattacharyya, 2018; Lample et al., 2016; Yuan, Lu, and Wu, 2017; Hammerton, 2003b; Chiu and Nichols, 2016; Chen et al., 2019; Habibi et al., 2017), the unary potentials $U(X_k, Y_k)$ are given by the last layer of the neural encoder. The purpose is then to find a label sequence Y which maximizes $P(Y|X)$ with respect to the parameters of the neural network and to the pairwise potentials which are learnt as well.

Thus, CRFs model the dependencies between labels and through a transition matrix that stores the information learned from the labels dependencies. As the impossible transitions are known, this transition matrix can be initialized in a manner that respects the diagram of impossible transitions shown in Figure 4.1. The pairwise potentials with high negative value in log space correspond to a probability of transition close to 0. By enforcing the values of these pairwise potentials it is possible to make certain transitions impossible.

CRF are often used in sequence labeling tasks like Named Entity Recognition (NER) (Saha, Saha, and Bhattacharyya, 2018; Lample et al., 2016; Hammerton, 2003b; Chiu and Nichols, 2016; Chen et al., 2019; Habibi et al., 2017). CRFs are also used in slot filling tasks (Saha, Saha, and Bhattacharyya, 2018) to build structured knowledge bases usable for semantic-based information retrieval. They are exploited in vision applications as well, for instance, for semantic segmentation (Zheng et al., 2015).

As the ability of deep neural networks to encode information is high, Neural Random Fields were introduced. A CRF is placed on top of a deep neural network

to take advantage of the high-quality extracted features (Peng, Bo, and Xu, 2009). For text mining, CRF are usually used with recurrent networks: Long Short Term Memory (LSTM) networks (Hammerton, 2003b; Chiu and Nichols, 2016) or Gated Recurrent Unit (GRU) networks (Cho et al., 2014). Recurrent networks (Hochreiter and Schmidhuber, 1997b) are known to be efficient for language processing as they allow information to be transmitted throughout the encoding of a sequence via a memory vector.

With the arrival of pre-trained encoders, which perform better than recurrent neural networks in NLP tasks, the trend (Li, Zhang, and Zhou, 2020; Yu et al., 2019; Xu et al., 2021) is to associate a pre-trained encoder (BERT (Devlin et al., 2018), XLNet (Yang et al., 2019b), etc.) with a CRF. An architecture with a pre-trained encoder and a CRF is chosen.

A limitation of the classical CRF is the lack of flexibility on the pairwise potentials. The transition matrix is unique regardless of the grammatical structure of the sequence under study. Approaches were developed in vision applications to generate pairwise potentials from Convolutional Neural Networks (Vemulapalli et al., 2016; Chen et al., 2018) but no approaches tackled the integration of syntactic information in pairwise potentials for text mining. Nevertheless, for a NER task and especially for TRIZ parameter mining, the position and the grammatical class of the words have an influence on the labels. The evaluation and action parameters follow syntactic patterns which are dependant on the patents' domain and writer. This explains the failure of pattern-based methods for their extraction. However, a few sentence constructions occur regularly, such as "The modification of this + AP + allows to improve + EP". Taking into account the syntactic structures can therefore bring additional information on the possible labels. Indeed, in the previous example, even without having read the EP, we can easily make the hypothesis that there will be an EP after the infinitive verb "to improve". That is why we will propose new CRF architectures which aim at taking into account the syntactic specificity of sentences containing TRIZ parameters.

4.2 Matrix-based Syntactic CRF

A first family of approaches is based on the hypothesis that instead of a single transition matrix it would be useful to have several and to make a choice with respect to the syntax of the sentence. The idea is to learn via transition matrices all possible situations where evaluation and action parameters appear.

Part Of Speech (POS) tagging is a very common task in NLP which aims to associate each word with its grammatical class. POS tagging being a popular task, very efficient models already exist : Irie et al., 2019; Brown et al., 2020; Radford et al., 2019; Melis, Kočiský, and Blunsom, 2020. Therefore, no models are re-trained for this part. In order not to increase the inference time, a relatively fast tagger is chosen from the spacy library. Indeed, a very high tagging quality does not appear necessary to recognize common syntactic structures.

Two different approaches are introduced to take into account the POS tagging information in the classification of tokens. The transition matrix gives the transition probabilities from the L_t label assigned to token t to L_{t+1} , label of token $t + 1$. In a classical CRF, this matrix is unique. A new matrix structure is presented in this section to model label dependencies while taking into account syntactic information. The first configuration takes into account the label of the considered token as well as the tokens which directly precede and follow the considered token. Three POS

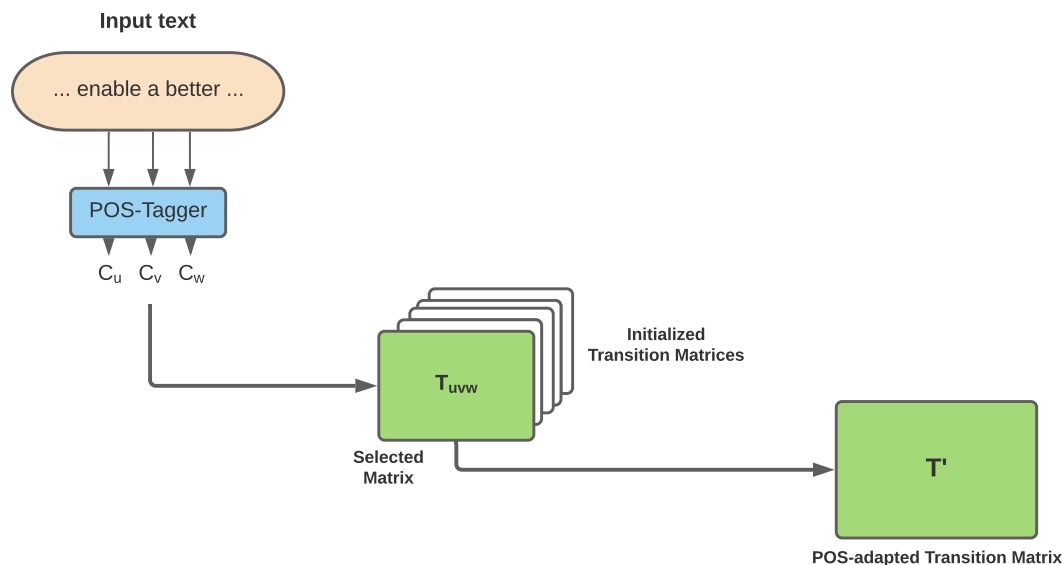


FIGURE 4.2: Multiple indexed transition matrices for POS information integration

tags are therefore used. The second configuration takes into account the label of the considered token as well as the two preceding and following labels. In this case, five POS tags are exploited.

4.2.1 SynCRF-matrix

This first approach (Figure 4.2), assumes that the transition matrix should be dependent on the syntactic structure. Indeed, if a token is found in a common grammatical structure for an evaluation parameter, for instance, the probability of transition to the EP label should be increased. Conversely, if the structure has little chance of being associated with an evaluation parameter this transition probability should decrease. A transition matrix is then initialized for each possible configuration of part-of-speech tags. For each label prediction, only the matrix corresponding to the series of three or five tags (depending on the configuration) is used. A tensor of dimension $(N_P, N_P, N_P, N_C, N_C)$ or $(N_P, N_P, N_P, N_P, N_P, N_C, N_C)$ with N_P the number of Part of Speech classes and N_C the number of labels for parameters mining task is used to index the transition matrices. In the example presented in Figure 4.2, we assume that a series of three POS labels C_u, C_v, C_w carry the syntactic information. This series of tags will directly correspond to a transition matrix at position (u,v,w) in the tensor presented above.

4.2.2 SynCRF-point

This second approach (Figure 4.3) aims at reducing the number of transition matrices. The principle is to initialize a constant (and small) number N of transition matrices T and to create a pointing mechanism towards the most adapted transition matrix from the series of parts of speech returned by the tagger. Thus, a few parameters are added and the transition matrices model the most emblematic cases only for the transition. The first step is the encoding of the combination of parts of speech tags. An encoding matrix E is therefore introduced. Hadamard products between

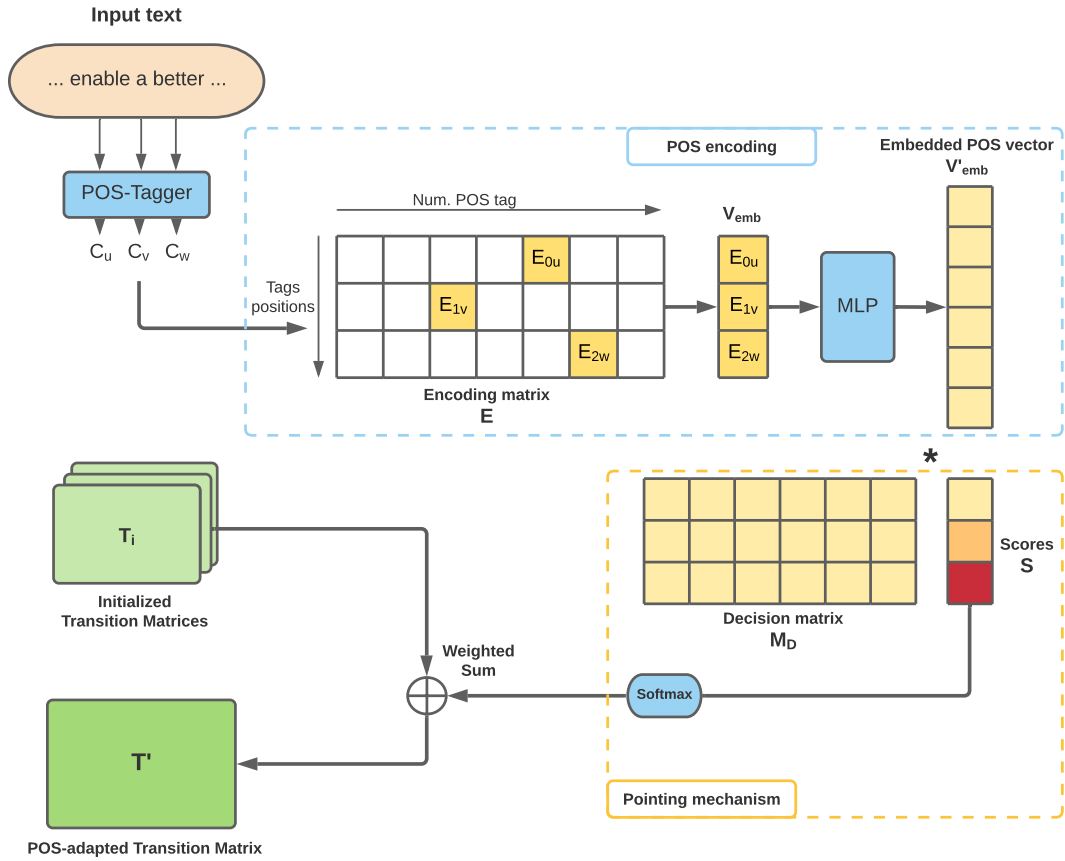


FIGURE 4.3: Pointing mechanism to integrate POS information in label prediction

the tags' one hot matrices (one hot vector for the POS tag with an additional dimension related to the position in the tag sequence (0,1,2) or (0,1,2,3,4)) and the encoding matrix allows the creation of an embedding V_{emb} containing the information on the tags and their position:

$$V_{emb} = \sum_j \sum_i E \odot \delta_i \delta_{j=tag_i}^T \quad (4.3)$$

with i the position in the tag sequence (from 0 to 2 if three tags are used for instance), j the POS class and tag_i the POS class of i^{th} tag.

V_{emb} then passes through a fully-connected neural network (FC):

$$V'_{emb} = FC(V_{emb}) \quad (4.4)$$

The product between a decision matrix M_D and the embedding V'_{emb} is then performed (Equation 4.5). Line i of M_D represents the "kind" of embedding that should be processed with transition matrix T_i . The product of M_D and V'_{emb} is therefore a scalar product between the targeted embeddings (meaning the embeddings meant to be processed by one of the initialized transition matrices) and the actual embedding. The product results in a vector S containing the scores associated to each transition matrix. A maximum score at position i means that the i^{th} targeted embedding is the closest to the actual embedding V'_{emb} . It implies that V'_{emb} should be processed with the i^{th} transition matrix.

$$S = M_D V'_{emb} \quad (4.5)$$

In order for the gradient to be back-propagated through all the transition matrices' parameters, M_D and E , the choice of the new transition matrix T' is modeled by a weighted sum of the matrices by the score vector (after application of a softmax function):

$$S' = \text{Softmax}(S) \quad (4.6)$$

$$T' = \sum_{i=0}^N S'_i T_i \quad (4.7)$$

with T_i the i^{th} initialized transition matrix, S'_i the score associated to matrix T_i and N the number of transition matrices chosen by the user.

The transition matrix T' will thus be unique for each sequence of tags but it will be very close to one of the existing matrices thanks to the application of the softmax function on the scores which puts the emphasis on the chosen matrix.

4.3 Potential-based Syntactic CRF

A second family of models called Potential-based Syntactic CRF is proposed. While Matrix-based Syntactic CRFs generate potentials linked to each other because they are derived from a transition matrix initialized at the beginning of the training, Potential-based approaches generate potentials, still based on the grammatical structure of the sentence, but independent from each other. This allows for greater adaptability according to the situations encountered.

4.3.1 SynCRF-pos: Part Of Speech-based Syntactic CRF

SynCRF-pos, shown in Figure 4.4, consists of two main parts: the encoding of parts of speech and the generation of pairwise potentials contained in the CRF's transition matrix. An encoding matrix E is introduced to make the transition between parts of speech and a numerical vector containing the information on the syntactic structure of the sentence. Sequences of five parts of speech are encoded (to simplify Figure 4.4, only three tags are considered). We, therefore, make the assumption that the label of a token is only influenced by the two preceding and following tokens. The one-hot-vectors, associated with the part of speech tags, allow selecting in E the parameters contained in the encoded vector V_{emb} . A Hadamard product is performed between the tags' one-hot matrix (one-hot vector for each of the POS tags concatenated relatively to their position in the tag sequence (0,1,2,3,4) and the encoding matrix E):

$$V_{emb} = \sum_j \sum_i E \odot \delta_i \delta_{j=tag_i}^T \quad (4.8)$$

with i the position in the tag sequence (from 0 to 2 if three tags are used for instance), j the index of the POS class (u, v, w in Figure 4.4) and tag_i the POS class of i^{th} tag.

V_{emb} is then upsampled via a fully-connected layer of neurons to give V'_{emb} :

$$V'_{emb} = FC(V_{emb}). \quad (4.9)$$

V'_{emb} is then used as an input for a neural network allowing the generation of these pairwise potentials. Several types of neural networks are implemented and compared in this approach: a fully connected 2-layer network and two recurrent GRU-type networks. The fully connected network directly integrates the syntactic

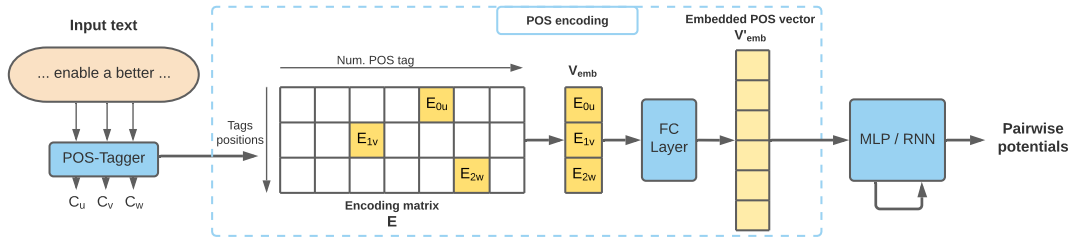


FIGURE 4.4: SynCRF-pos architecture for POS-adapted pairwise potentials generation

information contained in the encoded vector into a new transition matrix. On the other hand, the goal of the recurrent networks is to integrate a longer-term memory of the CRF and to emulate potentials that are not only dependent on the previous label but also on the preceding ones. Two configurations of recurrent networks are implemented. The first one aims at giving more weight to the last label than to the previous ones. V'_{emb} is thus aggregated to the memory vector (i.e. the hidden units, V_{hidden}) before generating the transition potentials using a fully-connected layer. The memory vector is then updated using V'_{emb} :

$$P_{i,j} = FC(V_{hidden}, V'_{emb}) \quad (4.10)$$

$$V_{hidden} = GRU_{update}(V'_{emb}) \quad (4.11)$$

with $P_{i,j}$ the pairwise potentials, FC a fully connected neuron layer, V_{hidden} the GRU's hidden units and GRU_{update} the hidden units' update function.

In the second configuration, the memory vector is first updated with V'_{emb} and then the pairwise potentials are computed from the new memory vector as follows:

$$V_{hidden} = GRU_{update}(V'_{emb}) \quad (4.12)$$

$$P_{i,j} = FC(V_{hidden}). \quad (4.13)$$

4.3.2 SynCRF-context: context-based Syntactic CRF

By adding a CRF on top of an encoder one takes advantage of the contextual representations of the tokens (Figure 4.5). Masked language models, due to their training, integrate rich syntactic information. It is, therefore, worth investigating generating the pairwise potentials of the Conditional Random Field from these contextual representations instead of using a part of speech tagging process. Additionally, the parts of speech tagging process adds computational complexity. A neural network computes the potentials given the token representations. Three different configurations are implemented for this neural network. A 1-layer and 2-layer fully connected neural networks are tested along with a recurrent neural network. A 1-cell GRU network is used. The purpose of this last configuration is to build a direct link between the generated pairwise potentials to improve consistency in label sequences. The token representation V_{rep} is fed into fully connected layer FC_0 to compute V'_{rep} (Eq.4.14). V'_{rep} along with the recurrent network hidden units V_{hidden} are then fed into a fully connected layer FC_1 to give the output pairwise potentials (Eq.4.15). The hidden units are finally updated using the input representation V_{rep} (Eq.4.16). The memory cell is therefore used to keep track of the input representations sequence while the feed-forward networks FC_0 and FC_1 are extracting the relevant features to predict

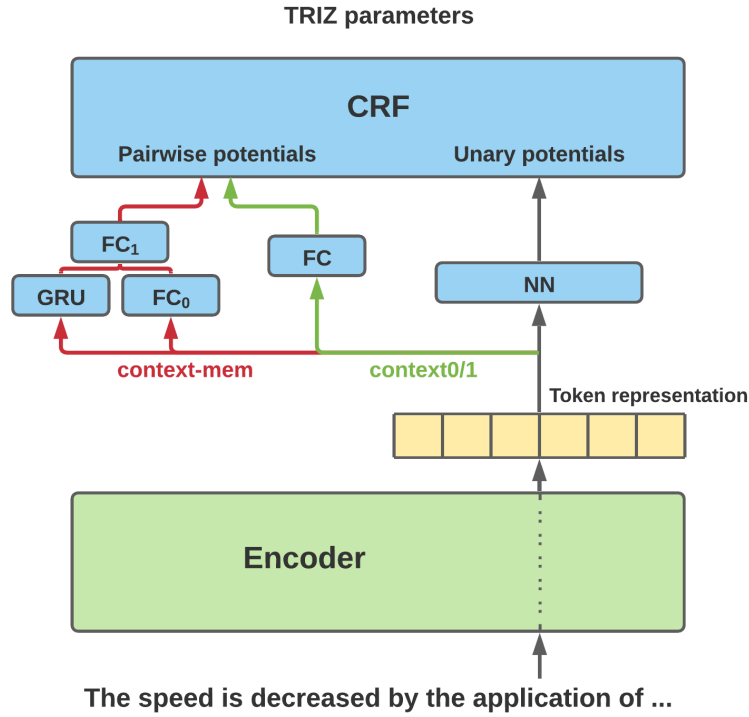


FIGURE 4.5: SynCRF-context architecture

the pairwise potentials as follows:

$$V'_{rep} = FC_0(V_{rep}) \quad (4.14)$$

$$P_{i,j} = FC_1(V'_{rep}, V_{hidden}) \quad (4.15)$$

$$V_{hidden} = GRU_{update}(V_{rep}). \quad (4.16)$$

The generation of "contextual" potentials is thus made possible by adding a minimum of parameters while remaining end-to-end trainable.

4.4 Dataset and training

Pre-trained encoders are designed to work well in domains suffering from data deficiency. TRIZ domain and patent analysis are especially concerned by the lack of labeled data as the labeling process is tedious and can only be performed by experts. A dataset of 1100 labeled patents was created and made available¹. It contains about 9000 labeled TRIZ parameters from abstracts, state-of-the-art, and claims parts of patents. Patents come from the United States Patent Trademark Office (USPTO). They were selected to cover all known technical domains (using CPC-IPC classification). A few statistics on this dataset can be found in Table 4.1. The dataset is clearly unbalanced with five times more EPs than APs but this actually does not constitute a major problem as the targets are EPs (for the contradiction) and APs are just additional (and not necessary) knowledge.

An example of a labeled sentence is given below:

"Thus, the size of the barrier must be closely matched to the size of the orifice to ensure that there are no gaps between the carrier and the panel member."

¹The dataset can be downloaded with [this link](#).

<i>Patents</i>	<i>EP – EP/doc</i>	<i>AP – AP/doc</i>	<i>Avg. words_{EP}</i>	<i>Avg. words_{AP}</i>
1093	8719-7.98	6604-1.51	3.79	3.29

TABLE 4.1: Details on the parameter dataset

The *size of the barrier* is labeled as an action parameter (AP) while *no gaps between the carrier and the panel member* is labeled as an evaluation parameter (EP). In the labeling policy, the parameters were defined as follows: an evaluation parameter is a parameter that measures the performance of a system, an action parameter is a parameter that can be modified and that influences one or more evaluation parameters. Verbs referring to changes in parameters (increase, decrease, etc.) are not included in the annotations. Two types of EP, EP+ and EP-, are defined to reflect either the positive or negative evolution of a parameter, or its positive or negative aspect (for example, a cost will fundamentally be a negative parameter). However, in this work, we do not consider the evolution of evaluation parameters and EP+ and EP- are aggregated in a single class EP. EPs are most often nominal groups (volume, power output, etc.) but verbal expressions can be annotated if no noun or nominal group can correctly describe the parameter. For example, "prevent fluid from entering the engine" will be annotated as it refers to sealing without the possibility of annotating a nominal group referring more directly to "sealing".

Syntactic CRFs are trained using gradient back-propagation. The additional fully connected layers on top of the encoder and the CRF are fully trained on the patent dataset while the pre-trained encoder is fine-tuned with a decreasing learning rate to avoid overfitting. The base learning rate is set to 3e-5 for the encoder and 1e-3 for the decoding part (Conditional Random Field or Fully connected layer for the *Baseline* model). The decoder has a higher learning rate as it has to be learned from scratch. A step learning rate decay is implemented. After the first epoch, the encoder learning rate is decreased to 6e-6 and then 3e-6 after the second epoch. Adam optimizer is used with a batch size of 16. The training is performed on a RTX2080Ti.

4.5 Experiments and results

4.5.1 Metrics and baseline approaches

Classification metrics are used to evaluate the models (Precision, Recall, F1-score). The accuracy is considered as not relevant to compare the models for this task. 4-fold cross-validation is performed.

Berdyugina and Cavallucci, 2022b is the only state-of-the-art approach to tackle parameter mining. This approach is based on a cause-effect framework. As the Action Parameters can influence the Evaluation Parameters, they are seen as causes of a change in an EP. The EPs are, therefore, seen as effects. It was trained on a cause-effect dataset.

To be able to compare with models using our data and measure the impact of our new syntactic CRF we, therefore, introduce several baselines. BERTDevlin et al., 2018 and XLNetYang et al., 2019b pre-trained encoders are used in SynCRF. They are both among the most widely used encoders. They are considered as our first baseline approaches. We add a simple classification layer with a fully connected layer on top of the encoders to mine parameters.

SynCRF is a neural random field (neural encoder with CRF). Thus we also consider neural random fields to have a fairer comparison with SynCRF. A CRFLafferty,

McCallum, and Pereira, 2001 is placed on top of both of these neural encoders to build two well-known models: BERT-CRFSun et al., 2022 and XLNet-CRFChai et al., 2022.

To highlight the impact of the transition constraints, we introduce two baseline approaches which are basically BERT-CRF and XLNet-CRF with the constraints shown on Figure 4.1 called BERT-CRF-cs and XLNet-CRF-cs.

4.5.2 Results

Table 4.2 shows the results of matrix-based SynCRF approaches with BERT encoder. SynCRF-matrix and SynCRF-point refer to the approaches described in Part 4.2.1 and Part 4.2.2. The configurations which take into account three POS labels sequences are marked with indice 1 in Table 4.2. The configurations working with five POS labels sequences are marked with an indice 2.

Table 4.3 gathers the results associated with Potential-based SynCRF based on BERT neural encoder while Table 4.4 contains the results associated with Potential-based SynCRF based on XLNet encoding. SynCRF-pos relates to the models using parts of speech (Part 4.3.1). *mem* and *mem-o* refer to the variation on the recurrent models described in 4.3.1. *mem* is the model described with Eq. 4.10 and 4.11 while *mem-o* refers to Eq. 4.12 and 4.13. SynCRF-context relates to the models using token contextual representations to generate pairwise potentials (Part 4.3.2). The number behind *context* indicates which configuration described in Part 4.3.2 is used. SynCRF-context-mem relates to the SynCRF-context variant with the memory cell.

The *cs* suffix indicates that probabilities of forbidden transitions are manually set to 0.

Table 4.5 finally compares the best SynCRF configuration versus the state of the art and baselines approaches.

E and *A* suffixes in the metrics refer to Evaluation Parameters (EP) and to Action Parameters (AP).

SynCRF-matrix

Overall, the results of SynCRF-matrix are quite weak compared to the other approaches. We notice that the variants with constraints on the transitions do much better. We observe, for example, an increase on the F1-score for the EPs of about 25% and of almost 45% for the APs with the variant 1 which takes into account 3 POS tags. The same orders of magnitude are visible with variant 2.

The best SynCRF-matrix variant is SynCRF-matrix1-cs which is better for all the metrics used. The loss is however slightly higher than that of SynCRF-matrix2-cs.

These relatively low performances can be explained by the large number of initialized matrices. Some of them are rarely used and thus, cannot be learned correctly with the backpropagation. This is why the SynCRF-point approach was developed.

Model	Loss	TP _E	Prec _E	Rec _E	F1 _E	Supp _E	TP _A	Prec _A	Rec _A	F1 _A	Supp _A
SynCRF-matrix1	0.299	14572	32.4	41.9	36.5	34868	744	19.1	11.3	14.1	6604
SynCRF-matrix1-cs	0.141	15468	47.3	44.5	45.9	34868	1308	40.1	19.9	26.5	6604
SynCRF-matrix2	0.293	14272	27.7	41.0	33.1	34868	772	13.1	11.6	12.1	6604
SynCRF-matrix2-cs	0.134	15012	46.6	43.2	44.8	34868	1064	35.6	16.2	22.2	6604
SynCRF-point1	0.390	16316	46.3	46.9	46.6	34868	1572	37.4	23.8	29.0	6604
SynCRF-point1-cs	0.149	15560	47.4	44.7	46.0	34868	1232	46.5	18.7	26.6	6604
SynCRF-point2	0.420	16136	46.0	46.4	46.2	34868	1524	37.8	23.1	28.7	6604
SynCRF-point2-cs	0.149	16080	48.6	46.1	47.3	34868	1320	43.9	20.0	27.5	6604

TABLE 4.2: Matrix-based SynCRF results with BERT encoding

Model	Loss	TP _E	Prec _E	Rec _E	F1 _E	Supp _E	TP _A	Prec _A	Rec _A	F1 _A	Supp _A
SynCRF-pos	0.164	16208	47.7	46.5	47.1	34868	1532	39.1	23.3	28.8	6604
SynCRF-pos-cs	0.162	16312	48.1	46.8	47.4	34868	1420	39.7	21.5	27.7	6604
SynCRF-pos-mem	0.134	15536	48.8	44.6	46.6	34868	1164	38.3	17.7	24.1	6604
SynCRF-pos-mem-cs	0.132	16368	49.4	47.0	48.2	34868	1340	38.0	20.3	26.5	6604
SynCRF-pos-mem-o	0.138	16140	48.3	46.3	47.3	34868	1272	41.5	19.2	25.9	6604
SynCRF-pos-mem-o-cs	0.138	16076	47.7	46.1	46.9	34868	1328	38.5	20.1	26.4	6604
SynCRF-context0	0.127	15920	50.5	45.6	47.9	34868	1472	40.3	22.3	28.6	6604
SynCRF-context1	0.125	16084	50.5	46.2	48.2	34868	1292	44.7	19.6	27.0	6604
SynCRF-context-mem	0.121	15560	52.2	44.6	48.1	34868	1480	47.2	22.3	29.9	6604

TABLE 4.3: Potential-based SynCRF results with BERT encoding

Model	Loss	TP _E	Prec _E	Rec _E	F1 _E	Supp. _E	TP _A	Prec _A	Rec _A	F1 _A	Supp. _A
SynCRF-pos	0.159	4182	50.6	47.6	49.0	35156	1696	37.9	25.0	29.9	6768
SynCRF-pos-cs	0.157	16196	50.9	46.1	48.4	35156	1640	42.1	24.3	30.4	6768
SynCRF-pos-mem	0.139	16096	51.5	45.9	48.4	35156	1276	37.3	18.9	24.9	6768
SynCRF-pos-mem-cs	0.134	16284	51.3	46.3	48.7	35156	1476	38.9	21.8	27.8	6768
SynCRF-pos-mem-o	0.291	4180	13.1	11.5	12.2	35156	340	9.2	5.0	6.5	6768
SynCRF-pos-mem-o-cs	0.134	16396	52.8	46.6	49.5	35156	1456	39.9	21.5	27.8	6768
SynCRF-context0	0.128	16680	53.2	47.4	50.2	35156	1532	49.1	22.6	30.8	6768
SynCRF-context1	0.122	16720	53.4	47.6	50.3	35156	1512	43.8	22.4	29.5	6768
SynCRF-context-mem	0.111	16752	52.6	47.7	50.0	35156	1628	43.7	24.1	31.0	6768

TABLE 4.4: Potential-based SynCRF results with XLNet encoding

Model	Loss	TP _E	Prec _E	Rec _E	F1 _E	Supp. _E	TP _A	Prec _A	Rec _A	F1 _A	Supp. _A
BERT (Devlin et al., 2018)	0.423	15076	31.6	43.3	36.5	34868	840	18.5	12.7	14.8	6604
BERT-CRF (Sun et al., 2022)	0.393	15504	37.8	44.5	40.9	34868	1136	26.7	17.2	20.6	6604
BERT-CRF-cs	0.137	15756	48.5	45.2	46.8	34868	1144	45.1	17.3	24.7	6604
XLNet (Yang et al., 2019b)	0.399	16592	38.0	47.2	42.1	35156	1272	26.1	18.8	21.7	6768
XLNet-CRF (Chai et al., 2022)	0.348	16888	43.7	48.1	45.8	35156	1260	31.2	18.6	23.2	6768
XLNet-CRF-cs	0.140	15276	48.7	43.6	45.9	35156	1056	42.3	15.6	21.6	6768
Berdugina and Cavallucci, 2022b	-	7548	11.0	21.5	14.6	35080	1916	2.5	28.9	4.5	6624
XLNet-SynCRF	0.111	16752	52.6	47.7	50.0	35156	1628	43.7	24.1	31.0	6768

TABLE 4.5: Comparison of SynCRF with the state of the art

SynCRF-point

This second matrix-based approach achieves much better results than SynCRF-matrix thanks to the limitation of the number of parameters learned via the pointing mechanism. As all the parameters are used for each forward or backward pass, the training is also supposed to be smoother. When SynCRF-matrix2 converges towards a local minimum SynCRF-pos2 converges towards a much better minimum. As the models learn transitions better, the constraints have logically less impact (>1% on the F1-scores).

The best model is SynCRF-point2-cs since, with comparable performance, it shows better accuracy than SynCRF-point2 for both EPs and APs. It also performs better than SynCRF-matrix variants which makes it the best matrix-based model.

SynCRF-pos

Results of the SynCRF-pos approach are very different according to the variants but the conclusions are the same for both neural encoders. We can see that adding constraints on the transitions allows to slightly decrease the loss (from 1% to 2% for SynCRF-pos-mem with BERT and XLNet encoder for instance). It also improves precision and recall by about 1% for EPs and 3% for APs. The addition of constraints to SynCRF thus allows constant but relatively limited improvements in the results.

Concerning the architecture, we highlight the relevance of adding temporal information on the previous pairwise potentials with a recurrent network. Indeed, we observe a decrease of about 20% in the loss between the non-recurrent SynCRF-pos models and the recurrent SynCRF-pos-mem models, whatever the encoder. On the metrics, we observe an increase in precision but a decrease in the recall, which keeps the F1 score at the same level. As precision is the most important metric in our case to avoid undermining bad contradictions the best SynCRF-pos model seems to be SynCRF-pos-mem-cs.

SynCRF-context

Using the richer tokens' representations of the encoder as a source of syntactic information shows, compared to the explicit syntactic information-based models (SynCRF-pos), a significant improvement in the results (Table 4.4). The loss decreases by about 10% between the best SynCRF-pos model and the best SynCRF-context model. The metrics are also positively impacted. The accuracy increases by about 6% with BERT versus 1% with XLNet for the EPs and by about 14% for the APs. The recall is relatively constant so it leads to an improvement in the F1 score.

The variant with the memory cell appears to be the best model in terms of loss and AP metrics while its performance on EP is as consistent as SynCRF-context0 and SynCRF-context1. SynCRF-context approaches also show overall better results than SynCRF-pos in terms of loss and metrics. This syntactic information also minimizes the impact of arbitrary constraints on certain transitions as these are learned by the network that generates the pairwise potentials. They outperform all constrained models without any external action on the pairwise potentials.

Comparison with the state of the art

Potential-based SynCRF globally perform better than Matrix-based SynCRF. Moreover, Matrix-based SynCRF necessitate to choose the number of parameters to encode the transitions (with the matrices) where Potential-based approaches do not

suffer from this limitation. SynCRF-context-mem is therefore the best SynCRF model. Table 4.5 compares SynCRF-context-mem, with the state-of-the-art approaches and baselines introduced. The contribution of a traditional CRF (BERT-CRF, XLNet-CRF) in the extraction of TRIZ parameters is visible in the results with a decrease of about 10% of the loss and of 4-5% of the F1-score for EP and AP compared to the encoders alone.

The addition of constraints on forbidden transitions (BERT-CRF-cs and XLNet-CRF-cs) has a strong positive impact on the loss value compared to the BERT-CRF and XLNet-CRF models (-60%) but the impact on the metrics is not constant depending on the encoder and the parameters' type. The precision is the only metric that is always improved by 5 to 10% with the additional constraints on the CRF. We, therefore, highlight that the interest in a traditional CRF is felt above all when one is aware of certain forbidden transitions which can be managed by imposing the values of the associated pairwise potentials. This impact is also much higher on a classical CRF than on our SynCRF.

The model of Berdyugina and Cavallucci, 2022b shows relatively weak performance compared to other models. The cause-effects framework does seem to fit well with the parameters because the recall is relatively high. It shows, for instance, the best recall for APs but the precision is extremely low so it is clear that there are a lot of false positives with this methodology and we cannot rely on it to extract contradiction parameters.

SynCRF largely outperforms all these approaches. Indeed, it shows consistent performance with both encoders. The loss is three times slower than encoders only and encoder+CRF architectures. The improvement on the metrics is massive especially for APs with a 25% improvement on the F1 score compared to the best baseline but also for APs with a 7% improvement on the F1 score. The precision is the most improved metric for EPs which is exactly what we are looking for. Thus, we demonstrate that adding syntactic information to generate pairwise potentials in a Conditional Random Field is very valuable, especially in tasks where labels are strongly linked to syntax like in TRIZ contradiction modeling.

4.5.3 Discussion

In terms of the used metrics, the values may appear low compared to traditional classification problems. It is nevertheless logical when one goes deeper in analyzing the extracted parameters. In theory, a parameter (EP or AP) is composed of several words for example *weight of a moving object*. Patents are written by non-TRIZ experts and the formulation of parameters is therefore not as straightforward. For example, in patent US07010885-20060314, the original TRIZ parameter *Loss of time* is described as *response time for completion of the service, application, and/or function*. In this case, even if the full sentence was labeled it would also make sense to only extract *response time* even though it gives less detail on the specificity of the parameter in the patent's context. In this case, if the model only labels *response time* as EP it would lead to a substantial decrease in the metrics. That's why, despite the low metric values, the models are still reliable. An example is given below with the patent US09131753-20150915, which is not part of the dataset. The original sentences to label, chosen by the summarization model introduced in Chapter 3, are the following: *In general, walking sticks assist physically challenged persons and hikers in walking by bearing a portion of the body weight and thus reducing the burden of the lower limbs. Being incapable of continuous illumination, the conventional walking stick not only lacks ease of use but also fails to ensure user safety*. The extracted EPs are: *bearing a portion of the body weight,*

burden of the lower limbs for the first sentence and *ease of use, safety* for the second one. No AP was mined. In this patent, concerning a walking stick with a light-emitting module, the purpose is, based on the extracted parameters, to develop a walking stick that eases the user's movement while increasing his safety.

In a second patent EP0489335 the chosen sentences are the following: *Such materials are advantageous in that they have high thermal conductivity and thus allow the melt of thermoplastic resin to cool rapidly and shorten the molding cycle time. The quick solidification of the melt combined with the limited flowability of the materials makes it difficult to achieve melt flow over a large area.* The extracted EPs for the first sentence are: *thermal conductivity, cool rapidly, shorten the molding cycle time.* For the second one *quick solidification of the melt, flowability, melt flow over a large area* were extracted as EP. Therefore, we understand that if the purpose is to shorten the molding cycle time by increasing the cooling speed, it is difficult to mold large objects. The patent describes an insulated mold.

With these two examples, we highlight the potential of our approach to mine parameters from patents to easily understand the underlying invention. We also highlight the difficulty of mining only two parameters for a contradiction while a sentence may contain five or six parameters. This gives room for further improvement in the contradiction mining process.

Conclusion

The mining of evaluation metrics (EPs) is necessary to identify contradictions. The action parameters can bring an additional level of analysis by indicating the action levers that have already been tried. The extraction of these TRIZ parameters is considered a Named Entity Recognition task.

The parameters of the evaluations and actions are often formed from several words. Dependencies between token-wise predictions are therefore modeled thanks to a Conditional Random Field (CRF). The CRF learns pairwise potentials that refer to the probability of transitions between labels. While these potentials are constant in the original implementation of the CRF, we have implemented a dependency between these potentials and the syntactic structure of the sentences. Indeed, TRIZ parameters often appear in close syntactic contexts. Among the approaches developed, the most efficient is the contextual approach which allows an absolute increase of the F1-score of more than 5% compared to a classical CRF. This approach can be extended to any Named Entity Recognition task as it doesn't depend on the tags nature. It is also able to efficiently spot forbidden transitions and does not add much parameters to the model (a few thousands depending on the number of tags). The real value of SynCRF approach compared to a classical CRF is the ability to better model the dependencies between labels. Thus, its implementation would have the best value when mining entities composed of several tokens.

Two different models have been developed for the contradictions and the parameters. To build a complete model, these two approaches will have to be merged. This is the purpose of the approaches introduced in the next chapter.

Chapter 5

Multi-task model for contradictory parameters and solution concept mining

Contents

Introduction	67
5.1 Naïve model and SummaTRIZ demonstrator	68
5.1.1 Module-based naïve model	68
5.1.2 Case study with SummaTRIZ demonstrator	68
5.2 Multi-task model	72
5.2.1 Backbone and shared parameters	72
5.2.2 Abstractive training and TRIZ Evaluation Parameters-Aware Attention (TEP2A)	74
5.2.3 Validation	75
5.2.4 Ideas Demonstrator	76
5.2.5 Interpretation of Inventive Principles	77
5.3 Experiments	77
5.3.1 Comparison between Naïve and Multi-task models	77
5.3.2 Parameters attention and modified loss	83
Conclusion	83

Introduction

The previous chapters have shown the development of two distinct approaches for identifying sentence contradiction relationships and mining TRIZ parameters. These two models can now be merged into a single model for mining contradictory parameters in patents.

Two approaches are possible. The first one consists in placing the parameter module above the contradiction module. This is going to be the naïve model (Section 5.1). The second approach consists in building a unique model inspired by the two already validated models allowing the joint extraction of parameters and contradiction relations (Section 5.2).

5.1 Naïve model and SummaTRIZ demonstrator

This section focuses on the fusion of the two modules (contradiction and parameters) and the demonstrator which was built using this model. The demonstrator aims at building a mapping of a domain's contradictions. More details and a simple tutorial for the demonstrator can be found in Appendix A.

5.1.1 Module-based naïve model

The naïve model allows the extraction of a contradiction (i.e. two parameters, one parameter per part of the contradiction) from a patent in a three-steps process. A sentence is first extracted for each of the two parts of the contradiction using the summarization model presented in Section 3. The parameters from these two sentences are then mined using the CRF-based model presented in Section 4. The probability of having a contradiction in the document is also evaluated with a document classifier using the sentences representations of both chosen contradiction sentences. The complete model is shown in Figure 5.1. The output of the model is therefore the parameters contained in both parts of the contradictions. The parameter(s) contained in the first sentence is(are) supposed to be improved. The parameter(s) contained in the second sentence need(s) to be degraded when the first parameter(s) are being improved. Theoretically, there should be only one parameter per sentence, but there is sometimes more than one. This does not affect the process if we extend the notion of contradiction to a set of parameters that cannot be improved at the same time.

The model integrates the limitations of its two sub-modules. The summarization model was developed in order to process documents with a size of less than 1600 tokens, i.e. containing less than 50 sentences to match the length of dataset documents. If the input document is longer than this limit, it is separated into chunks of length less than or equal to 1600 tokens. The chunks are then processed independently. When choosing the "parts of the contradictions" sentences after the sentence classifiers, all the scores of all sentences in the document are taken into account. This ensures that the contradiction search is done on the whole document but it leads to a degradation in the performance. However, as it is rare for the state of the art to contain more than 50 sentences, this degradation has not been studied.

A demonstrator called SummaTRIZ¹ has been developed to implement this full contradiction mining model. It is able to search the patent database indexed on one of our servers or load user's patents.

5.1.2 Case study with SummaTRIZ demonstrator

To illustrate the possible benefits of a TRIZ parameter mining process we have applied this naïve model to a corpus of patents dealing with LCD screens using our demonstrator. A selection of sentences containing the parameters of each contradiction is performed via the summarization neural network. For instance, on patent US08749511-20140610, the selected sentences are the following: *In the market of flat panel displays, liquid crystal display LCD devices have obtained a relatively dominant position because of the characteristics such as smaller volume, lighter weight, lower power consumption, lower radiation, etc.. The conventional LCD device requires two additional switching elements in each pixel region, and therefore its structure becomes complicated and the number of manufacturing processes increases.*

¹<https://summatriz.inventivedesign.unistra.fr/>

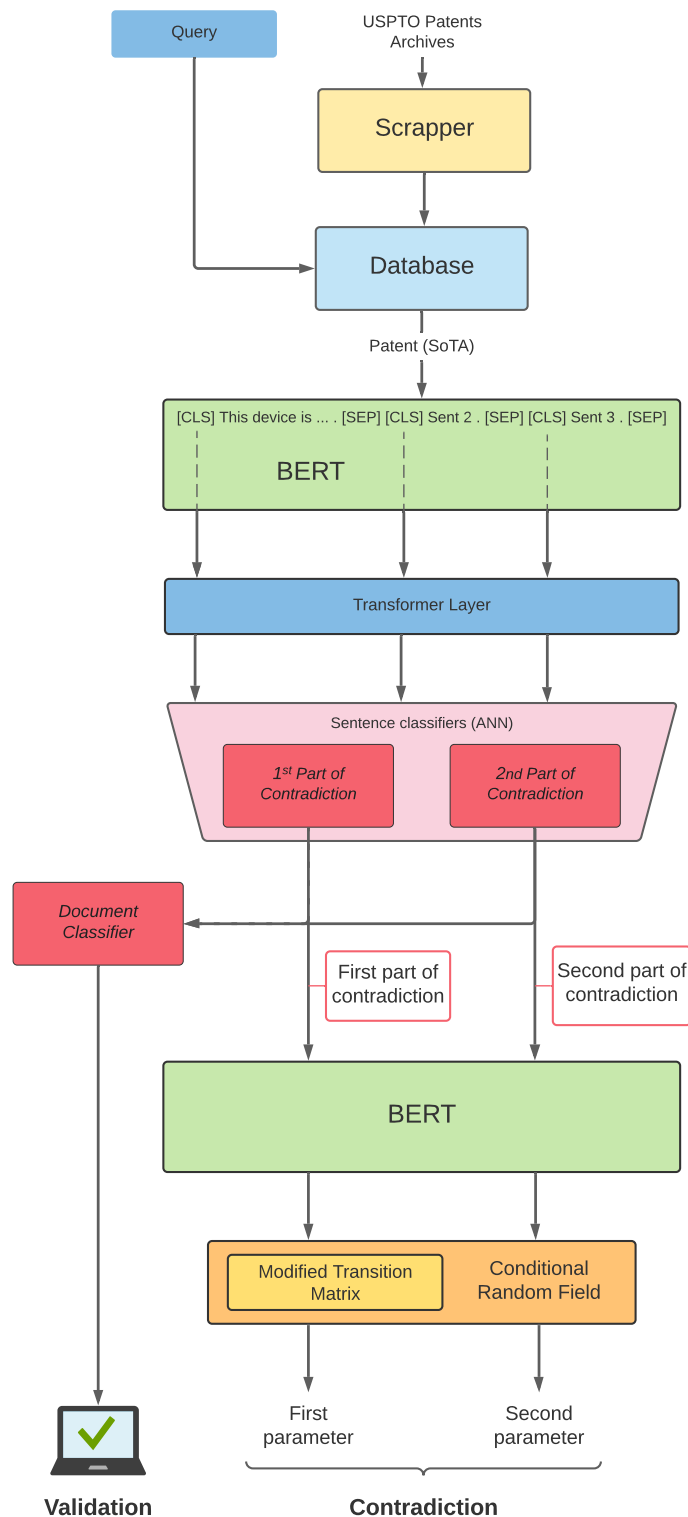


FIGURE 5.1: Naïve approach

SynCRF is then used to mine the evaluation parameters (we will not use the action parameters in this study as they give information on partial solutions which are not of interest here). On the patent example, it gives: *volume, lighter weight, power consumption, lower radiation* for the improved parameters (first sentence) and *structure become complicated, number of manufacture processes* for the degraded parameters

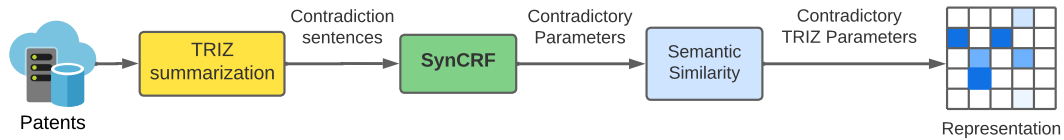


FIGURE 5.2: Process used in the case study to represent LCD screens main contradictions/research topics

(second sentences).

Finally, each extracted parameter is mapped to an original TRIZ parameter among the 39 (see Table 2.3). This mapping allows similar parameters to be grouped and is no more and no less than a clustering of the mined EPs. This similarity exploits a pre-trained Transformer of type MPNet (Song et al., 2020) on a dataset of one million pairs of semantically similar sentences. Each extracted PE is transformed into a simple sentence of the type: "\$PE must be improved" and is compared to the 39 sentences formed from the 39 TRIZ parameters. Thus, each extracted contradiction is reduced to a contradiction between TRIZ parameters and can be represented in a matrix of constant size 39*39. The mapped TRIZ parameters for patent US08749511-20140610 are *volume of moving object*, *weight of stationary object*, *loss of energy*, *temperature* and *shape*, *ease of manufacture* for both improved and degraded parameters. We then make the hypothesis that each improved parameter is in contradiction with each of the degraded parameters. By extending this application to 100 patents on LCD screens a heatmap (Figure 5.3) is created to represent the central issues of the LCD domain.

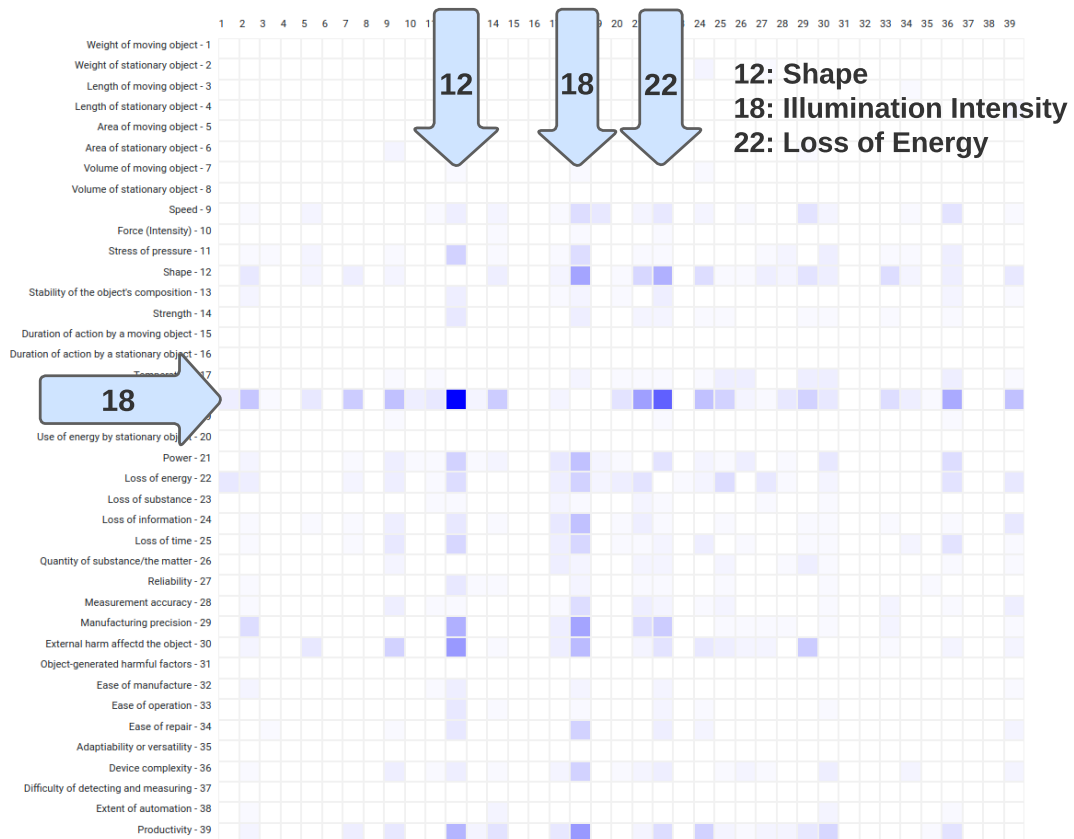


FIGURE 5.3: TRIZ contradictions occurrences in LCD screens SoTA

The original TRIZ parameters are found on the two axes of the matrix whose order corresponds to that of the Table 2.3. The colour scale corresponds to the frequency of conflicting parameter pairs in these 100 patents.

The first observation allows us to identify the rows and columns that are the most coloured. We find a main row corresponding to parameter 18 (Illumination intensity) and two main columns corresponding to parameters 12 (Shape) and 18 again. Thus, in a few seconds we can identify the parameters that will count in the system.

In order to identify the TRIZ contradictions related to LCD screens, we must look at the content of the cells and in particular at the content of the most colored cells. The two darkest cells are located on line 12. The first one corresponds to a contradiction between Illumination Intensity and Shape. Table 5.1 shows examples of mined parameters that have been mapped onto these two TRIZ parameters. It is first noticeable that the mined parameters are indeed evaluation parameters. Several contradictions are, indeed, present in this cell, given the mined parameters. On the one hand, the thickness, thin profile, and flatness cannot be reduced without degrading the quality of the image, and the same applies to the shape (wide viewing angle). The second darkest cell concerns a potential contradiction between parameters 18 (Illumination Intensity) and 22 (Loss of energy). At first sight, this contradiction seems trivial. We can check the parameters that were mined (Table 5.2). They correspond rather well to the TRIZ parameters. However, it can be noticed that in both examples there are parameters which would require a more thorough verification, such as "sizes" which is very vague, "brightness of backlight" which might not correspond to the studied contradictions, or "free radiation". Nevertheless, we show here in this trivial example that our approach allows us to be effective at mining contradiction parameters. We notice that sometimes verbs are present when they are not necessary but this remains anecdotal since the mapping on the corresponding TRIZ parameters always works. Our approach allows us to retrieve the main research trends and the limits of the state of the art very quickly based on the TRIZ parameters and contradictions.

TRIZ parameters	18. Illumination Intensity	12. Shape
Mined parameters	improve image quality light emission efficiency contrast of the LCDs devices brightness of backlight color difference	flatness wide viewing angle sizes thickness thin profile

TABLE 5.1: EPs mapped on TRIZ EPs for 18/12 contradiction

TRIZ param.	18. Illumination Intensity	22. Loss of energy
Mined param.	improve image quality uneven brightness image quality display effect of the LCD can be reduced high aperture ratio	flatness free radiation power consumption low power consumption consumes less power

TABLE 5.2: EPs mapped on TRIZ EPs for 18/22 contradiction

5.2 Multi-task model

When the two modules are simply stacked, this creates a model with a large number of parameters (about 400M) and the two tasks are independent while the parameters and contradictions are intimately linked. The choice of sentences should depend on the results of parameters mining. Moreover, the extractive summarization achieves good performance but it implies to modify the usual functioning of the encoder by adding classification tokens and an additional layer to link all sentences. The pre-trained encoder must therefore be pre-trained again after adding these parameters. It is then fine-tuned with TRIZ data. The training process is therefore long and complex. Abstractive summarization does not have these shortcomings, but the integrity of the information is no longer guaranteed if the sentences are imperfectly paraphrased. To circumvent this problem, we propose to divert an abstractive summarization model (Figure 5.4) to extractive summarization, i.e. we teach the abstractive network to copy sentences from the input document. The dataset for extractive summarization can, thus, be reused. Moreover, this model allows the mining of parameters, contradictions and inventive principles to be performed at the same time without adding parameters. The choice of the sentences in contradiction will be influenced by the results of the parameters mining. Finally, we propose a real classification of the extracted contradictions instead of the classification of documents operated in the summary module presented in the chapter 3 which does not allow us to conclude on the quality of the extraction but only on the presence of a contradiction in the document.

5.2.1 Backbone and shared parameters

BART (Lewis et al., 2020a) is chosen as the basic component of the multi-task model. It is a proven encoder-decoder for automatic summarization. A model derived from BART, called LED (Beltagy, Peters, and Cohan, 2020) is utilized. Local attention gives more importance to the close context during the analysis of a content and limits the number of computations. It is thus possible to process texts of up to 16384 tokens. This is compliant with contradiction mining as the interesting information is rarely very far away, even if the text is long. The basic model has 160M parameters, i.e. almost three times less than the naive model presented in the section 5.1.

The embeddings at the output of the encoder helps mining the evaluation parameters, the action parameters and the inventive principles. For the parameters, the dataset used to train the parameter module of chapter 4 is reused. For the inventive principles, a dataset proposed by Boufeloussen and Cavallucci, 2021 is chosen. This dataset is composed of abstracts and full-texts of biology articles. The words corresponding to each of the inventive principles are extracted from the texts. The classification of tokens (i.e. Named Entity Recognition) is therefore possible. An example of annotation with the article of Gaschk, Frère, and Clemente, 2019 is provided below:

*On the ground, the locomotion of koalas resembled a **combination of marsupial behaviours and primate-like mechanics**. [...] These results suggest that during ground locomotion, **they use marsupial-like strategies but alternate to primate-like strategies when moving amongst branches, maximising stability in these environments**. The locomotion strategies of koalas provide key insights into an independent evolutionary branch for an arboreal specialist, highlighting how locomotor strategies can convergently evolve between distant lineages.*

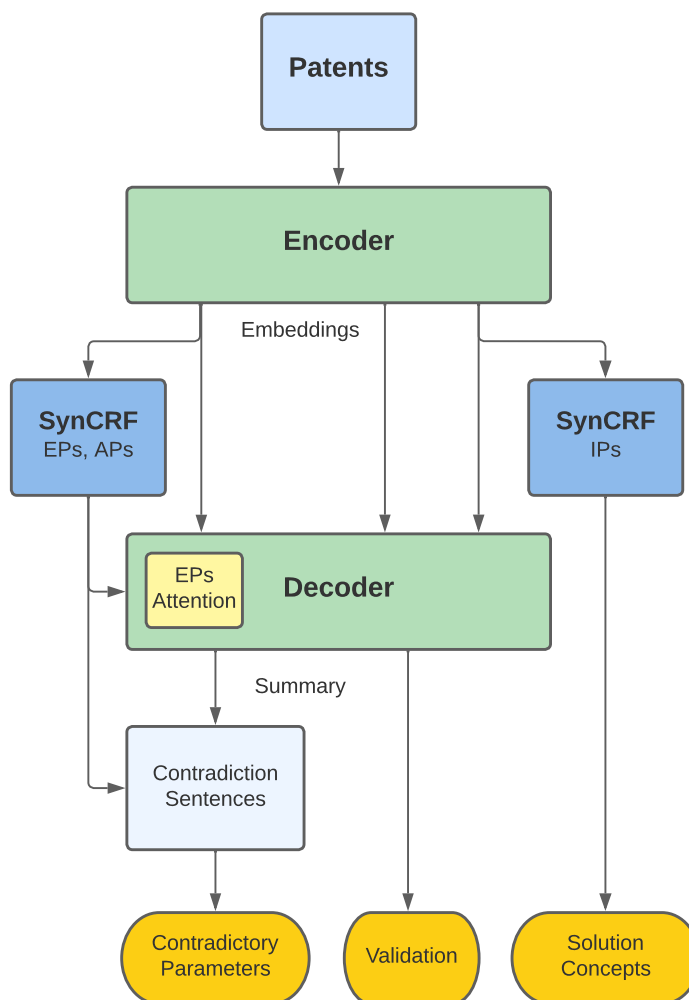


FIGURE 5.4: Multi-task model

The first expression in bold is classified as *Combining*, principle 5, which seems fairly obvious. The second expression is much less obvious with principle 3 *Local Quality*. The classification of inventive principles seems more difficult and will depend heavily on the personal appreciation of the annotators. The dataset includes about 2000 articles whose abstracts have been labeled. The dataset is highly unbalanced. For example, there are 500 examples for *Change of physical and chemical parameters* and *Feedback* while there are only 5 examples for *Homogeneity* and 7 for *Thermal expansion* (all details can be found in Figure 5.5). To avoid this problem, only two classes (Inventive Principle and a rejection class) are used. This still allow us to identify interesting passages concerning the solutions.

SynCRF is exploited to bring coherence in the predictions at the token level as shown in Figure 5.4. As the network is designed to handle long sequences, the computation time due to the estimation of the likelihoods of the sequences of labels at the level of the CRFs is very high (complexity $O(L)$ with L the length of the sequence) because these computations are not parallelizable. The chosen solution is to split the input sequences into smaller sequences (100 tokens maximum). The splitting is done by sentence. Thus the CRFs are trained on dozens of sentences separately. The embeddings from the encoder are therefore split in the same way. The entire tag sequences are then reconstructed as output from the CRFs.

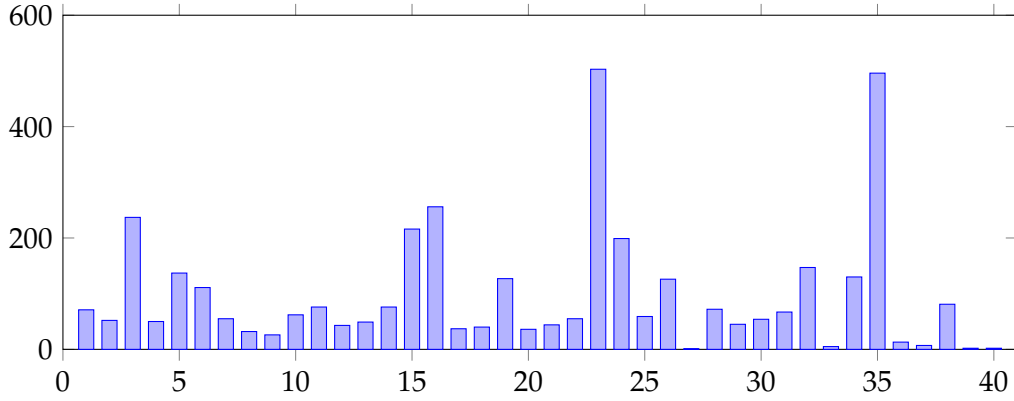


FIGURE 5.5: Distribution of inventive principles in the dataset

5.2.2 Abstractive training and TRIZ Evaluation Parameters-Aware Attention (TEP2A)

As the embeddings from the encoder contain features allowing to qualify the parameters of evaluations and actions, we can assume that the summary already integrates this knowledge. Nevertheless, to try to bring an explicit information on the presence or not of parameters, we introduce in the attention mechanism (Vaswani et al., 2017), during the decoding, an information on the presence of evaluation parameters. Moreover, abstractive training for TRIZ contradictions poses several problems, especially for computing loss. These aspects will also be discussed.

TEP2A

The attention allows us to choose in each layer which representations and therefore which words are the most important to build the representations in the following layer. The idea is therefore to simply modify the attention values in the decoder so that words close to the parameters are taken into account more.

Let us go back to the formula associated with the attention mechanism:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (5.1)$$

with Q , K , V the Queries (what information the layer needs), the Keys and Values (what information it actually has), d a rescale factor.

In transformer decoders, encoder-decoder attention are couples with self-attention (when Q , K and V are the same). In the encoder-decoder attention V and K are the encoder embeddings while Q come from the decoder. It can therefore be written as follows:

$$Attention_{E-D}(Emb, Q) = softmax\left(\frac{QEmb^T}{\sqrt{d}}\right)Emb \quad (5.2)$$

with Emb the encoder embeddings.

We add up the parameters attention coefficients C_p with the existing coefficients:

$$Attention_{E-D}(Emb, Q) = softmax\left(\frac{QEmb^T + C_p}{\sqrt{d}}\right)Emb \quad (5.3)$$

The parameters' attention coefficients are computed using parameters unary potential and convolutional layers. The purpose is to identify the patterns corresponding to sentences with parameters. 3 successive layers are used with a kernel size of 7. Kernel size was chosen to be big enough to analyze an entire sentence and small enough not to take too much context into account.

Abstractive training and loss

Contradiction mining through abstractive summarization shows some limitations. The summarization dataset is built in such a way that all sentences containing the contradiction parameters are labeled. During the abstractive training the model is, thus, taught to generate sometimes redundant sentences. If the model generates non-redundant sentences, even though the contradiction could be correctly mined, the loss would be high. This is mitigated by the use of teacher forcing during training. Indeed, as the generation of the summary is self-regressive (each token is generated from the previous tokens), as soon as the generated summary diverges from the ground truth, measuring the loss and back-propagating the gradient does not make much sense and can create instability during learning. The solution is therefore to use a teacher, i.e. to generate the $(n + 1)^{th}$ token from the n^{th} tokens of the ground truth.

From this observation, we can deduce that abstractive learning boils down to two things: learning to copy the sentences of the input document for all the tokens apart from the beginnings and ends of sentences, and learning to manage the choice of sentences (for the tokens at the beginning and end of sentences). Learning to copy the input sentences is a trivial task. The real difficulty lies in the transitions between sentences and between parts of contradictions. To model the two parts of contradictions we introduce a special token $\langle \backslash s \rangle$ which corresponds to the end of document token. The summary thus follows the following form: *Sentence 0. Sentence 1. $\langle \backslash s \rangle$ Sentence 2. Sentence 3..*

As the network learns very quickly to copy the input sentences, the loss associated with the summary decreases rapidly until it becomes less than or equivalent to the other losses. Learning then slows down. To counter this effect, we introduce a mechanism of weights allocated to tokens according to their interest. Tokens near transitions will have a higher weight than others. The weights associated with the tokens are given by a sum of half-Gaussian means of the tokens at the beginning of sentences or special tokens and standard deviations 2. Thus the token of interest is given a maximum point and the next 4 tokens are given lighter weights:

$$C_p(x) = \sum_{j=0}^n \mathcal{N}(x|z_j, \sigma^2) \mathbb{1}_{x>z_j} + C_0 \quad (5.4)$$

with j the index of the separation, z_j the position of the separation j , σ the standard deviation set experimentally to 2 and C_0 a constant.

5.2.3 Validation

In chapter 4, we introduced a classification of documents to compensate for the absence of a validation of the identification of the contradiction. Indeed, it is easier to analyze a text a priori by estimating the probability of describing a contradiction than to validate an extracted contradiction in particular. In the first case, if several

sentences present problems and/or solutions, it is almost certain that there is a contradiction to solve. However, choosing the best sentences to describe this contradiction remains difficult. Without better options, document classification can still filter out a high number of uninteresting patents (Table 3.7). The interest of switching to an abstractive summary rather than an extractive one is the possibility of using the principle of teacher forcing of the summary to do contradiction classification and more simply a document classification. Indeed, by moving the classification problem from the encoder to the decoder, we no longer classify the presence or absence of a contradiction in the document but the presence or absence of a contradiction in the summary imposed by the teacher. Learning is done by back-propagation of the error. A global loss is introduced which is the weighted sum of all the losses corresponding to the different tasks. All weights are set to 1 by default. Having found that validation at the decoder level implied instability during training, an experimental weight of 0.025 is applied to the validation loss during training.

5.2.4 Ideas Demonstrator

The multi-task model was integrated into an existing software called Ideas. The purpose of the software is to guide a user from the definition of his problem to its resolution. The choice of the contradiction to solve is not trivial since several contradictions are usually present. It is therefore necessary to be able to choose which contradiction to solve according to the goal and the development cycle of the product. Contradictions are constructed from problems and partial solutions (solutions that improve some parameters but degrade others) in the form of graphs. An example of a human-built graph for a scooter is shown in Figure 5.6.

In order to build this graph, it is necessary to have a certain amount of knowledge on the subject, which is not always the case. This is why the automatic extraction of problems and contradictions from documents related to the subject allows to save a significant amount of time. The idea is to populate the graph with knowledge extracted by our multi-task model to make the construction of the problem graph more efficient and faster. An example of a graph constructed from 7 patents randomly selected among the results of a search with the keywords "scooter assembly" is shown in Figure 5.7. The patents used are as follows: US07464784, US07044488, US10023254, US06889788, US11203391, US10787218, US10723403.

Each sub-graph is given by the analysis of a patent. For the moment no link between problems and solutions from different patents is made. Only 4 sub-graphs appear because the graphs with a probability of contradiction lower than 0.5 are automatically eliminated in order not to overload the graph. The observation of the sub-graphs shows us two main things. The first is that by taking random patents based on the title, it is difficult to have a homogeneous corpus in terms of themes. We find at least one patent on electric scooters, one patent on standing scooters and one on a self-balancing scooter. This is also linked to the fact that the search terms "scooter assembly" were not discriminating enough. The second lesson is that the sub-graphs do look like contradictions and that in a few seconds we can already have parts of graphs that make sense with real problems related to a subject and that only need to be linked and completed. This avoids starting from a blank page and greatly accelerates the beginning of the analysis process. The parameters are not displayed on the graph in the current state of the software but will be in a future version. The links between the nodes are formed from the output summary of the model. When the sentences of the first or second part of the contradictions are consecutive in the

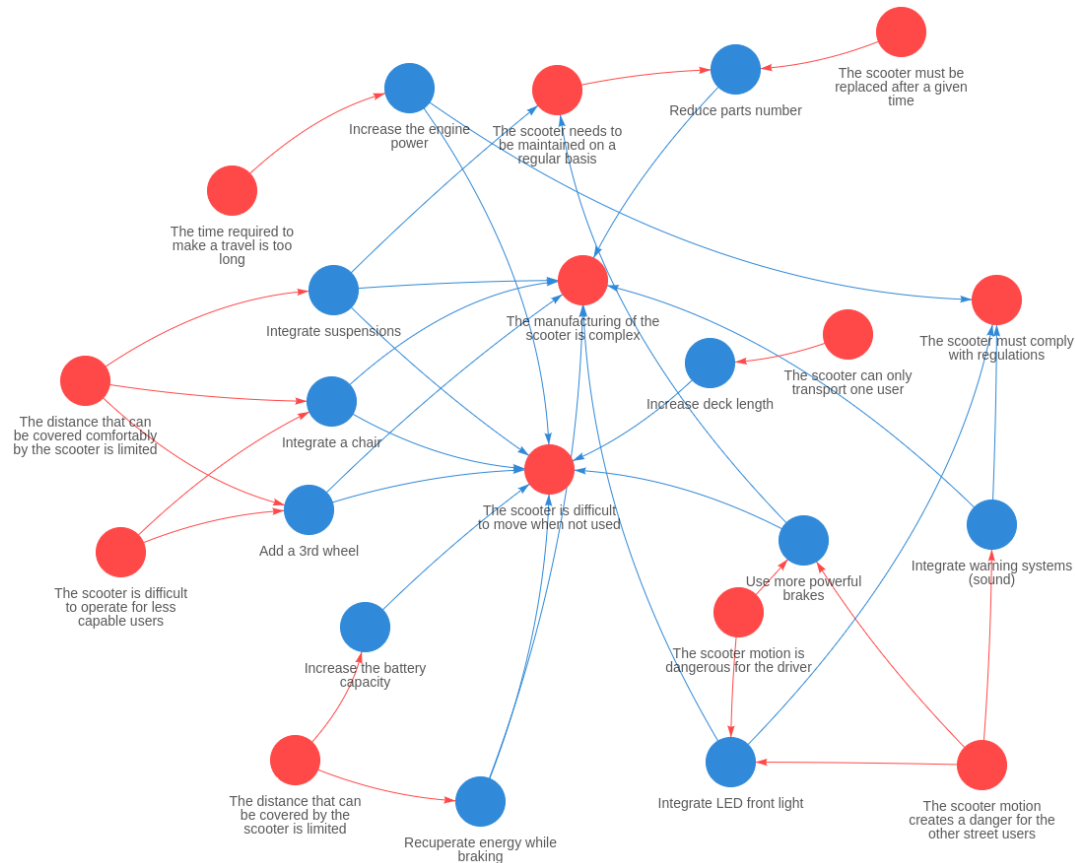


FIGURE 5.6: Example of problem graph within Ideas for a scooter

text, the nodes are placed in series (1st graph). Otherwise they are placed in parallel (2nd graph).

5.2.5 Interpretation of Inventive Principles

Some extractions using the inventive principle classifier are shown in Table 5.3. These phrases are supposed to refer to an inventive principle. At least two observations can be made considering these results. Firstly, the phrases are relatively long and often too long. For example "auto-balance driven vehicle" would be sufficient for the first example or "central wheel structure device" for the second example. Nevertheless, as the labeling examples often take the entire sentences, this is expected. Secondly, precision, at least on those examples, could be further improved as half of the results do not refer directly to inventive principles.

5.3 Experiments

5.3.1 Comparison between Naïve and Multi-task models

In this section we will compare both models in terms of quantitative performance on the datasets.

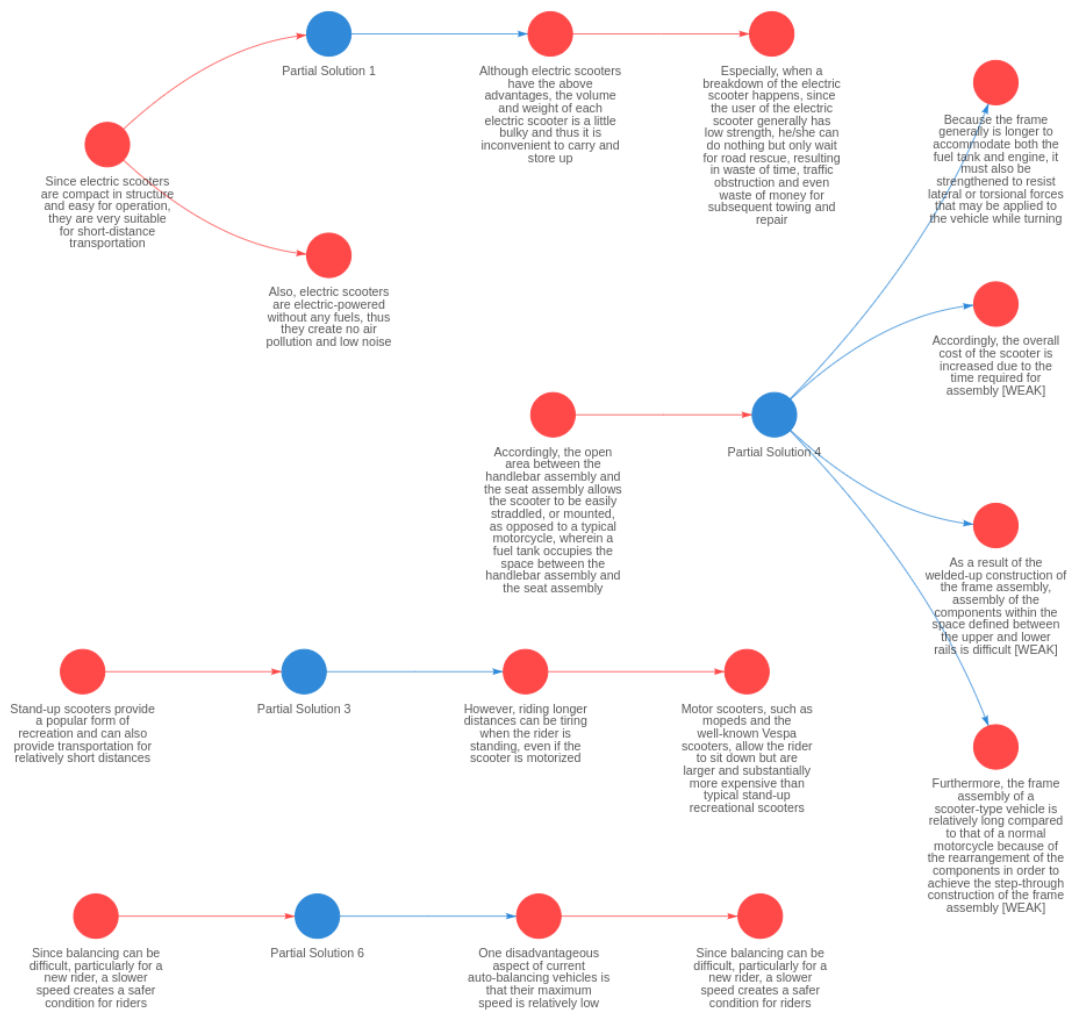


FIGURE 5.7: Example of problem graph built automatically

Labeled phrases	IP
cost-effective manner	-
a need exists to increase the speed at which an auto-balance driven vehicle may be safely operated, and to improve learning rate, increase ease of use, and enhance stability	Self-service (25)
when used with a central wheel structure device (ie, wheel structure located between the foot platforms), a scooter assembly enhances lateral stability, particularly when the vehicle is in motion	Equipotentiality (12)
it would permit a rider to carry an object more confidently, and to carry a heavier object (such as groceries or the like)	-
the provision of a scooter assembly would allow greater ease of use	-
If, however, a support or other structure can be provided that improves balance and/or reduces the probability of a rider falling off, then maximum speed may be increased	Anti-weight (8)
a slower speed creates a safer condition for riders	-

TABLE 5.3: Identified Inventive Principles in patent US10723403

Model	TP_{EP}	$Prec_{EP}$	$Recall_{EP}$	$F1_{EP}$	$Supp_{EP}$	TP_{AP}	$Prec_{AP}$	$Recall_{AP}$	$F1_{AP}$	$Supp_{AP}$
Naïve model	16752	0.53	0.48	0.50	35156	1628	0.44	0.24	0.31	6768
Multi-Task	19686	0.68	0.55	0.61	35461	1280	0.42	0.20	0.27	6478
Multi-Task _{doc}	19659	0.69	0.55	0.61	35461	1329	0.40	0.21	0.27	6478

TABLE 5.4: Multi-Task model vs Naïve model (EP/AP)

Model	$TP_{F/sum(1)}$	$TP_{S/sum(1)}$	$TP_{C/sum(1)}$	$TP_{F/sum}$	$TP_{S/sum}$	$TP_{C/sum}$	$Length_{F/sum}$	$Length_{S/sum}$	$Length_{C/sum}$
Naïve model	-	-	-	-	-	668	1	1	1
Multi-Task	736	1136	575	823	1350	736	1.4	2.0	2.0
Multi-Task _{doc}	747	1137	567	833	1370	741	1.4	2.0	2.0

TABLE 5.5: Multi-Task model vs Naïve model (Contradictions)

Model	$TP_{V/sum(1)}$	$Prec_{V/sum(1)}$	$Rec_{V/sum(1)}$	$F1_{V/sum(1)}$	$F1_{V/sum(1)}$	$TP_{V/sum}$	$Prec_{V/sum}$	$Rec_{V/sum}$	$F1_{V/sum}$	$Supp_{V}$
Naïve model	-	-	-	-	-	576	-	-	-	3200
Multi-Task	499	0.26	0.87	0.40	0.40	641	0.33	0.87	0.48	3200
Multi-Task _{doc}	464	0.25	0.82	0.38	0.38	618	0.33	0.83	0.47	3200

TABLE 5.6: Multi-Task model vs Naïve model (Validation)

Model	$Loss_{IP}$	TP_{IP}	$Prec_{IP}$	Rec_{IP}	$F1_{IP}$	$Supp_{IP}$
Multi-Task	0.07	33172	0.57	0.41	0.47	81440
Multi-Task _{doc}	0.07	35456	0.54	0.44	0.48	81440

TABLE 5.7: Multi-Task model (Solution Concepts / Inventive Principles)

The Tables 5.4-5.6 compare the naive model to the multi-task model for the identification of the parameters (EP, AP), the contradiction retrieval and the validation of the extraction. Due to the different natures of the trainings and thus of the metrics, only part of the performances can be compared. Nevertheless, this will be enough to demonstrate the impact of the multi-task model.

Table 5.4 is used to compare the mining performance for the evaluation and action parameters. The traditional classification metrics are used. We notice a clear improvement on the evaluation parameters (+11% in absolute value on the F1 and especially +15% on the precision). On the contrary, the performances of the action parameters mining are slightly worse with a reduction of 4% in absolute value on the F1 score. Since a contradiction, in its simplest form, can only show evaluation parameters, the action parameters are not a priority. We thus see here a clear improvement of the results for the parameters with the implementation of the multi-task model.

Table 5.5 attempts to compare the results in terms of contradiction mining. The naive model considers only one sentence per contradiction part while the multi-task model will sometimes consider several. The average lengths of the summaries are shown in the *Length* columns. The indices *F*, *S* and *C* refer to the first parts, second parts and contradictions as a whole. A priori, we do not know which sentence is the most important when there are several in the first or second part of a contradiction. The index *sum(1)* considers only the first sentences of the first and second part of contradiction from the obtained summary. For example, let us consider a summary with two sentences in the first part of the contradiction. If the first sentence of the summary is indeed part of the contradiction, it will count as one for $TP_{F/sum(1)}$ and $TP_{F/sum}$. If the second one only is part of the contradiction it will count as one for $TP_{F/sum}$ but not for $TP_{F/sum(1)}$. We can see that by keeping only the first sentences of the summary, there is a 21% decrease in the number of contradictions extracted. Therefore it seems important to keep the whole summary. With the whole summary, 736 contradictions out of 1600 are extracted. With the naive model, 668 contradictions are extracted. The multi-task model thus allows an improvement of 10% on the mining of the contradictions compared to the naive model.

Table 5.6 attempts to compare the number of contradiction retrievals validated by each of the two models. In the naive model, the validation is done with a document classification while in the multi-task model it is a summary-level classification. Traditional classification metrics are used. The purpose of this layer is to add a layer compared to Table 5.5. In table 5.5, only the summary is considered. In this table, both the summary and the score of the document/contradiction classifier are analyzed. If the contradiction is correctly mined in the summary but the score/contradiction probability is less than 0.5 it will count as 0 for both $TP_{V/sum(1)}$ and $TP_{V/sum}$ while it will count as 1 if the score is more than 0.5. We introduce a multi-task model with a document classification, called *Multi – Task_{doc}* for a more accurate comparison. We notice that the classification at the decoder level is slightly better for the recall (4% improvement) and thus on the F1-score but the improvement remains very small. Thus, for three extractions supposed to be contradictions, only one really is. The other metrics (parameters, contradictions) are similar between document classification and contradiction classification. A possible improvement would be a classification of the special token generated between the two parts of contradictions. Thus, the classification would really be a contradiction classification. Nevertheless, this token may not be generated if there is no contradiction or it could be generated several times.

Finally, for the inventive principles, the results are shown in Table 5.7. The performance of the model is equivalent for parameter mining and solution concept mining. Note that to increase the precision, sentences which do not contain an inventive principle have been added to the dataset. This addition was done at the expense of recall but what matters the most is the quality of the extracted contents. Nevertheless, we highlighted in Table 5.3 that this was still not precise enough.

Model	Loss	TP_{EP}	$Prec_{EP}$	$Recall_{EP}$	$F1_{EP}$	$Supp_{EP}$	TP_{AP}	$Prec_{AP}$	$Recall_{AP}$	$F1_{AP}$	$Supp_{AP}$
Multi-Task	0.158	19686	0.68	0.55	0.61	35461	1280	0.42	0.20	0.27	6478
Multi-Task $_{attn}$	0.158	20427	0.66	0.58	0.62	35461	1168	0.43	0.18	0.25	6478
Multi-Task $_{scores}$	0.158	19347	0.69	0.55	0.61	35461	1184	0.41	0.18	0.25	6478

TABLE 5.8: Multi-Task model variants (EP/AP)

Model	Loss	$TP_{F/sum(1)}$	$TP_{S/sum(1)}$	$TP_{C/sum(1)}$	$TP_{F/sum}$	$TP_{S/sum}$	$TP_{C/sum}$	$Length_{F/sum}$	$Length_{S/sum}$
Multi-Task	0.149	736	1136	575	823	1350	736	1.4	2.0
Multi-Task $_{attn}$	0.149	731	1127	565	715	1357	715	1.3	2.0
Multi-Task $_{scores}$	0.148	739	1128	561	824	1360	734	1.4	2.0

TABLE 5.9: Multi-Task variants (Contradictions)

Model	Loss	$TP_{V/sum(1)}$	$Prec_{V/sum(1)}$	$Rec_{V/sum(1)}$	$F1_{V/sum(1)}$	$TP_{V/sum}$	$Prec_{V/sum}$	$Rec_{V/sum}$	$F1_{V/sum}$	$Supp_{V}$
Multi-Task	0.536	499	0.26	0.87	0.40	641	0.33	0.87	0.48	3200
Multi-Task $_{attn}$	0.509	455	0.25	0.81	0.39	592	0.33	0.83	0.47	3200
Multi-Task $_{scores}$	0.548	496	0.26	0.88	0.40	646	0.33	0.88	0.48	3200

TABLE 5.10: Multi-Task variants (Validation)

Model	Loss	TP_{IP}	$Prec_{IP}$	Rec_{IP}	$F1_{IP}$	$Supp_{IP}$
Multi-Task	0.074	33172	0.57	0.41	0.47	81440
Multi-Task $_{attn}$	0.075	29580	0.54	0.18	0.25	81440
Multi-Task $_{scores}$	0.074	32306	0.54	0.40	0.46	81440

TABLE 5.11: Multi-Task variants (Solution Concepts / Inventive Principles)

5.3.2 Parameters attention and modified loss

To study the impact of the parameter-wise attention and of the redistribution of the weights given to the tokens for the abstractive training, additional experiments are conducted. The results are compiled in Tables 5.8-5.11.

The results show that the redistribution of the token weights for the abstractive loss computation has a very slight positive influence on the loss. However, the results are overall the same for all the metrics, which leads us not to opt for this choice in the default configuration of the multi-task model.

The conclusion is similar for the parameter-wise attention mechanism which even leads to a decrease in the main metrics related to contradictions. The only visible positive impact is a significant decrease in the loss associated with the validation (Table 5.10, -6%). However, despite this decrease in loss, the associated metrics are still below the levels of the other models metrics. This choice of architecture is therefore not retained.

Conclusion

Two complete contradiction mining models are presented in this chapter. The first one is based on a simple superposition of the contradiction and parameter mining modules presented in the previous chapters (called naïve model). The second one is a multi-task model allowing to learn jointly to mine parameters, contradictions and solution concepts (through inventive principles).

A multi-task TRIZ model allows to decrease the number of parameters (1/3), while building richer representations of the tokens, since integrating information on more different concepts (parameters, contradictions, inventive principles).

The multi-task model allows for an improvement in performance compared to the naive model. The F1 score for the mining of evaluation parameters increases by 15% and 10% more contradictions are extracted. While the contradiction mining model used in the naive model is derived from a semi-supervised training, the multi-task model achieves better results with a simple supervised training.

Explicitly taking into account the position of the parameters for the choice of the sentences in contradiction does not seem to have a positive effect. We can hypothesize that this is already the case implicitly with embeddings that include all information. Similarly, giving greater importance to tokens at the beginning of sentences for abstract training does not show a significant improvement in results.

An end-to-end model can build domain mappings in a few minutes and shape the R&Ds decisions to focus on high-value contradictions. It can also help minimizing the time to find the core contradiction to solve. Inventive principle mining should be improved as the false positives are still numerous. This could require re-annotating a dataset as the one used has shown its limits (in terms of interpretability of the labels).

Conclusion and future works

Contents

A work in line with the current era	85
Contributions	85
Perspectives	86

A work in line with the current era

The R&D process is still very much based on technological monitoring. To be able to identify a problem, or a contradiction, it is necessary to have a global vision of what already exists in the competition or in the field. This research is carried out by expert services in technology watch but the tools used are often simply keyword search engines. Indeed, it is for example unimaginable to sort documents by the type of problems they deal with by hand since this would require reading them all. This is not humanly possible. In the same way, when it comes to answering a problem, brainstorming or trial and error are approaches whose results are uncertain and which rely above all on the experience of those who use them.

With competition getting stronger and time constraints going hand in hand (Shirwaiker and Okudan, 2008; Jardim-Goncalves et al., 2011; Smirnov et al., 2013) it seems obvious to rely more and more on repeatable and efficient methods allowing to innovate and be ahead of competitors (Hao et al., 2019; Renjith, Park, and Kremer, 2020; Kusiak, 2016). This was the purpose of TRIZ when it was developed by Altshuller and it is still the purpose of IDM (Inventive Design Methodology). Standardized formulations of problems and solutions allowed Altshuller to build a pseudo-algorithm for inventors that almost assures them to be innovative. It is this algorithmic form that justifies attempts at automation (Liang and Tan, 2007; Ni, Samet, and Cavallucci, 2022; Wang et al., 2016).

With neural networks becoming more and more powerful, especially with the transformers architectures (Devlin et al., 2019; Yang et al., 2019a; Beltagy, Peters, and Cohan, 2020), automation is becoming one of the "hot" topics. In this thesis, our objective was to build models allowing a characterization of the problems and solutions of patents of all domains to identify problem/solution couples. These problem/solution pairs can then be exploited to facilitate the resolution of new problems. Problems are described in the form of domain-free contradictions. Inventive principles, i.e. the way the problem was solved, are used to describe solutions.

Contributions

The contributions are structured around the objective of characterizing problems and solutions.

In Chapter 4, a model for identifying contradictions in USPTO patents from automatic extractive summarization is developed. A BERT-based transformer (Devlin

et al., 2019) with a dual binary classification is first presented. This model takes advantage of transfer learning and an additional attention layer to model the links between sentences (here contradiction links in the sense of TRIZ). A semi-supervised learning with a Generative Adversarial Network allowed us to clearly improve the performances of the model. A document classification also eliminates patents that do not contain contradictions.

In Chapter 5, a Named Entity Recognition-like model for mining TRIZ parameters (Evaluation Parameters and Action Parameters) based on an XLNet-like encoder (Yang et al., 2019b) is developed. Since the syntactic structure of the sentences containing these parameters is more or less constant, an improved version of a Conditional Random Field, called SynCRF, based on the syntactic structure is developed. The pairwise potentials will be generated either from the Part of Speech tags or from the embeddings and will thus be modified at each iteration. In the initial version of the CRFs, these potentials were constant. This model shows performances unmatched by other state-of-the-art approaches.

In chapter 6, two complete models of contradiction extraction are proposed (sentences + parameters + validation). The first model simply consists in naively stacking the modules on top of each other. The second model is a true multi-task model trained on all tasks at the same time (to which the inventive principles have been added). The model is based on a backbone (Beltagy, Peters, and Cohan, 2020) on which several SynCRFs are placed (one for each type of token classification). The decoder generates an abstract summary. In order not to lose any information, the learned abstract summary consists in copying the "contradictory" sentences of the input text. The validation mechanism used is no longer a document classification but a real summary classification. Finally, the contribution of explicit information on the parameters during the generation of the summary is studied. The idea is to push the network to choose only sentences containing evaluation parameters for the TRIZ summary.

In addition to these contributions, we built two datasets used to validate the summary and parameter mining models.

Perspectives

The perspectives can be organized into two groups: application perspectives and methodological perspectives.

In terms of application, the logical consequence of this work is the construction of a database of problem/solution pairs. The problems are the contradictions solved by each patent and the solutions are the inventive principles identified in the descriptions or claims of these same patents. A database of solutions accessible via a search engine can be built and which, to each input contradiction, would associate possible solutions to this contradiction. This also opens the way to a generative model trained to write claims from the contradictions for example. The multi-task model identifies passages referring to inventive principles. If these passages can be matched with inventive principles, it will be possible to train a GPT-like model (Brown et al., 2020) with a Contradiction + Inventive Principle prompt to generate the corresponding claim. The advantage of such a technique is the limitation of the number of possible solutions. In fact, a limit can be set to one or two solution generations per inventive principle in order to guarantee small volumes of generated texts that can be analyzed by experts. However, the understanding of the inventive

principles seems, to date, an extremely difficult task as it requires a real understanding of the solution. To find the inventive principle, one must first extract the core of the solution (which has been attempted in the last chapter of this thesis). Then, it is necessary to interpret it, which most often requires an understanding of what the system is, of the interactions between the components of the system and of the implications of the solution on the system. This is an original task that would surely require the identification of the interactions between the elements in the system via, for example, a graph. The semantic similarity between the extracted textual elements and the inventive principles could also be a way but the important number of classes (40) and the very different forms that solutions inspired by these inventive principles can take are the main limitations of this approach.

One of the major drawbacks of the current contradiction mining approach is the number of parameters found in the sentences. While a contradiction is, in theory, formed by only two evaluation parameters, we regularly find 4 or 5. One way to choose the most interesting ones would be to segment the sentences according to the cause/consequence relationships in order to eliminate the parameters whose degradation is only a consequence of the degradation of another parameter. Another option would be to reconstruct a new dataset that really identifies the parameters of the contradiction. Instead of the two datasets developed for the two main tasks (summary, parameter mining), we would have a single dataset, with the sentences of the contradictions and the parameters of these sentences that are in contradiction. We could also take advantage of this to add the solutions for pairing with the inventive principles and thus have a complete dataset allowing, at choice, to develop decoupled approaches for each task or not.

As for the methodology, a first track to study would be reinforcement learning. Indeed, the current abstractive loss does not take into account redundant sentences. The model is therefore penalized if it does not generate all the sentences that contain the parameters of the contradiction, whereas, theoretically, one sentence for each part of the contradiction is sufficient. One could therefore imagine a reinforcement learning policy that only takes into account the presence of the parameters of the contradiction. Another possibility would be to use a multi-goal classification (Guzmán-rivera, Batra, and Kohli, 2012). In this case, all possible correct summaries are compared to the generated summary and only the smallest loss is taken into account (hindsight loss).

Adding explainability with a semantic graph would also be a possible option. Since a semantic graph represents very well the interactions between nodes, it could be interesting to explain contradiction relations (for example if an element leads to an improvement or a degradation of another element we come closer to the definition of a contradiction parameter). One can even imagine that embeddings of nodes learned thanks to graph neural networks (GNN (Sperduti and Starita, 1997)), for instance, would be more able to represent the interactions between the trizian entities that are the evaluation and action parameters. GNNs can be seen as a generalization of transformers allowing to process data of any nature (image, text, structure of a molecule, interactions on a social network). As soon as the data can be represented as a graph with nodes and edges, GNNs can be used. For example, for an image, the nodes will be the pixels and the edges will be the adjacency relations between the pixels. In a text, the words will be the nodes. In a molecule, it will be the atoms. Each node and edge will be represented by a vector. A GNN can perform node-wise predictions, edge-wise predictions and global graph prediction. It allows to iteratively build meaningful representations for edges and nodes. The fact that node representations are intrinsically linked to edge representations makes

these networks particularly suitable for hybrid tasks such as those considered here with the modeling of contradiction relations (edges) between parameters (nodes).

The last step to fully automate the solution generation process is the interpretation of a potential solution from a different domain. This interpretation can be done implicitly with a generative model as discussed above but one can also think in terms of projections into domain spaces. If it is possible to build a representation of a solution one could for example see it as a sample drawn from a particular distribution depending on the domain and the idea behind the solution i.e. the inventive principle. An engineer will have difficulty interpreting a solution from a domain in which he is not an expert. The question is then: can we project the solution in the engineer's domain of expertise or find an equivalence between an existing solution in his domain and the target solution? Optimal transport (OT) could be an answer. Recently, a model based on OT was able to establish word translations through embedding matching without any labeled examples (Alvarez-Melis and Jaakkola, 2018). One could therefore imagine in the same way constructing links between solutions from two different domains. However, the problem would be complex since we are not sure that a solution exists to the optimal transport problem and that indeed there is a possible link between solutions from two different domains. In the same vein, inspired by style transfers with GANs in the case of paintings, we could try a projection of the solution in a target domain. The idea is to generate texts that would look like real solutions (first loss of the discriminant) but that would also look like a text of the target domain (second loss of the discriminant). Thus we would create a solution from scratch with, a priori, the vocabulary of the targeted domain. However, unlike optimal transport where the goal is to link existing inventions, with a GAN there is no guarantee that the generated solution would actually make sense.

Appendix A

Demonstrators

Contents

A.1 Patent scrapper and database	89
A.2 Softwares and computing	90
A.3 Summatriz (Naïve model, domain mapping)	90
A.3.1 Load patents or make a query to retrieve patents	90
A.3.2 Analyze patents	91
A.3.3 Optional: correct the results	91
A.4 Ideas (Multi-task model, problem graph generation)	92
A.4.1 Starting a project	92
A.4.2 Selecting the data	92
A.4.3 Creating the problem graph	92

In this chapter we share further details about the two demonstrators integrating the approaches developed in this thesis. Summatriz (available here ¹) allows operating the naive model and building a representation of a domain in terms of contradictions. SummaTRIZ was developed solely for this thesis. Ideas demonstrator (available here ²) is the flagship demonstrator of our team and aims at building an innovation assistant for engineers. Ideas integrates the latest version of the multi-task model which is used to build a graph of problems/contradictions involving a particular system.

A.1 Patent scrapper and database

For the purpose of this thesis, a large number of patents must be available. This is useful for demonstrators but also to build large datasets with the first trained models. Indeed, after understanding the problem-solution pairs from the patents, the goal is to build an idea generator trained on millions of patents.

A scrapping algorithm has been developed to download the entirety of the weekly US patent archives from the US Patent Office (USPTO) website. The patents are then extracted from the XML archives and each part is identified using keywords. In classical patent databases only the abstract, descriptions and claims are identified. Here, as the developed approaches use the state-of-the-art part, the breakdown must be finer to separate the state of the art from the description.

To build a patent search engine, an Elasticsearch database based on the Lucene library is used. All the fields extracted from the patents (title, reference, inventors, applicants, abstract, keywords) are indexed in the database which makes multi-criteria searches possible.

¹<https://summatriz.inventivedesign.unistra.fr/>

²<https://ideas.inventivedesign.unistra.fr/>

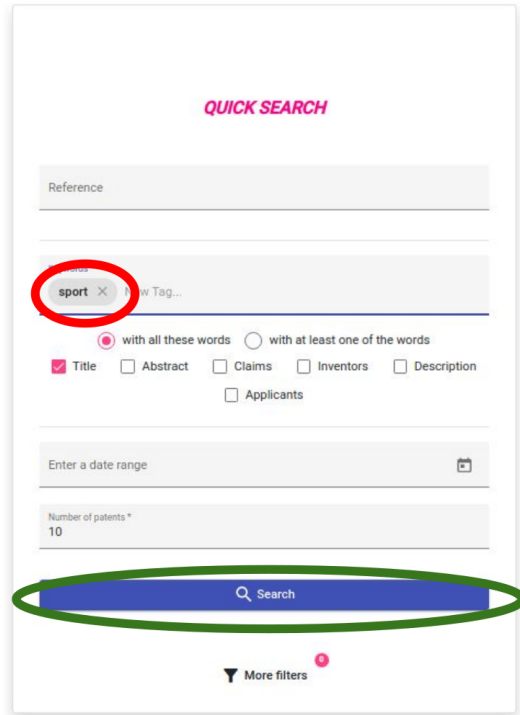


FIGURE A.1: SummaTRIZ: searching for patents

A.2 Softwares and computing

In order to be responsive to user requests coming from the demonstrators, the approaches are generally deployed as apis in containers. In this thesis, a computing server has been installed and maintained to accelerate the inference of neural-based models. To share resources (gpus, cpus, memory), slurm, a workload manager used in computing centers and supercomputers, is deployed. It allows distributing the computational loads on the gpus and avoid that several apis use the same resources at the same time. However slurm does not handle http requests and only supports job submission i.e. script execution. An api called *slurm_mgr* has therefore been developed to interface between the software and slurm. The softwares send requests to *slurm_mgr* which submits the corresponding jobs and provides the input data to the script. At the end of the job the script sends the output data to *slurm_mgr* which sends it back to the software.

A.3 Summatriz (Naïve model, domain mapping)

SummaTRIZ was built to demonstrate the effectiveness of data mining approaches on the key concepts of TRIZ (contradictions and parameters). The software is accessible to everyone with the username *test* and password *test*.

A.3.1 Load patents or make a query to retrieve patents

To load patents (for example the patents provided here ³, use the upload button (in red on A.2). To make a query, enter one or several keywords (in red on A.1) and use the Search button (in green on A.1)

³<https://drive.google.com/file/d/14F18J3wg0W3V3hLnOx8TsC5QWhEbamp1/view?usp=sharing>

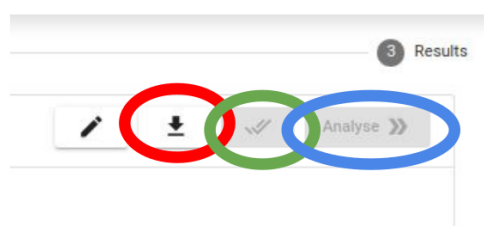


FIGURE A.2: SummaTRIZ: uploading patents and validation

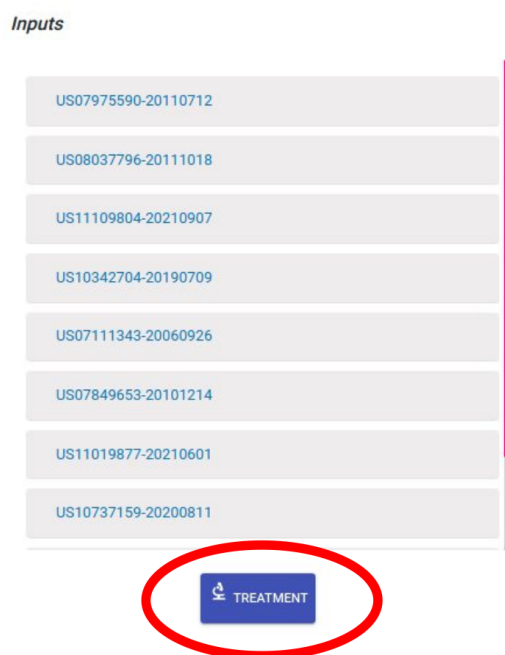


FIGURE A.3: SummaTRIZ: analyzing data

A.3.2 Analyze patents

Validate your selection of patents using the button highlighted in green in A.2. Then click on *Analyse* (in blue on A.2). The “Treatment” button shown on Figure A.3 will start the PaTRIZ model to mine contradictions.

After a few seconds/minutes, the results will appear. On the left (in green on A.4) are the sentences which have the most chances to be the first part of contradiction and on the right (in red on A.4) the second part of contradictions. Below each sentence are shown the extracted parameters. By switching to the *Results* page (Button on the upper right of Figure A.4), the representation of the domain can be displayed in a matrix format.

A.3.3 Optional: correct the results

Some buttons (in blue on A.4) are available if the choice of sentences is not correct or if parameters are missing. Using these buttons the user can modify the sentences or add/remove parameters before exporting results (using the “Export” upper button, note: this is still under development).

A.4 Ideas (Multi-task model, problem graph generation)

The multi-task model is integrated into Ideas, which is the official demonstrator of the team's applications for industrial partners. The objective is to help building a problem graph that will make the authors of the study choose the best contradiction to solve. This short tutorial concerns only the part of the software involving the multi-task model.

A.4.1 Starting a project

After starting the creation of a project, you must choose a Finder project as shown in Figure 1. One can then choose the name of the project and invite other participants or a team already created before. A description of the project can also be written. Once the project is created, a new page allows you to set up the schedule and assign tasks to the participants (Figure A.6).

A.4.2 Selecting the data

After validating the first two steps, a window opens to select the patents to be analyzed in the context of the study (Figure A.7). The database created for this thesis is utilized (Figure A.8).

A.4.3 Creating the problem graph

After validating the selection of patents, their content can be extracted by pressing *Extract* (Figure A.9). By validating the extracted elements, the graph appears containing the contradictions identified in the selected patents. Each graph corresponds to a patent. The patents with a probability of contradiction lower than 0.5 are ignored.

The screenshot displays the SummaTRIZ software interface. At the top, there are two tabs: 'Analyze' (active) and 'Results'. Below the tabs are 'Export' and 'Save' buttons. The main area shows two TRIZ ideas, each with a description, parameters, and a 'Replace' button. The left idea is highlighted with a green oval, and the right idea is highlighted with a red oval. A network diagram connects the two ideas. At the bottom, there is a 'Replace' button and a slider for 'probability_contradiction threshold'.

Left Idea (Green Oval):
 As a result, dashboard panels can be heavy and difficult to transport, assemble and adjust.
 parameters: PE : heavy + PE : difficult to transport assemble and adjust +

Right Idea (Red Oval):
 problems associated with the above portable containment wall systems may include damage to the anchoring rods, such as bending or breakage, difficult dowel removal, and instability of support legs due to their removable and rotational disposition on the panel.
 parameters: PE : damage to the anchoring rods + PE : bending or breakage + PE : difficult dowel removal + PE : instability +

Bottom Right Idea (Red Oval):
 During periods of nonuse, the support leg can be rotated into the recess in a storage position, permitting easy stacking and storage of the panels with the support legs neatly disposed in the rear recesses.
 parameters: PE : easy stacking and storage +

Bottom Left Idea (Blue Oval):
 In addition, dashboard systems are designed to permanently affix to the ground and do not have bracing options that allow the dashboards to be portable.
 parameters: PE : portable +

Network Diagram: A diagram showing connections between the ideas. Red lines connect the top of the left idea to the top of the right idea, and the top of the bottom right idea to the top of the bottom left idea. A blue circle highlights the 'Replace' button for the bottom left idea.

Interface Elements: 'Export', 'Save', 'Replace', 'Adjust the probability_contradiction threshold' slider.

FIGURE A.4: SummaTRIZ: results

× Create a Project

Cover:

Type:

Name:

Team: Reuse Create

Team Name:

Participants:

Description

Normal **B** *I* U

Tell your coworkers a bit about your project...

FIGURE A.5: Ideas: creating a project

1 Define Your Project


Description

No description


Tasks

FIGURE A.6: Ideas: defining a project


Projects / Démo Finder Multi-task





Démo Finder Multi-task

Type: Finder
Created: 24 days ago by [Guillaume Guarino](#)
Rights: ● You **can** edit this project
Task Progress: 0%
Last Activity: 23 days ago
Active users: 

2 Extract Outside Sources

Patents
7 

News Articles
0 


Papers
1 

[Patents](#) [News Articles](#) [Research Papers](#) [Papers \(Legacy\)](#)

Show only selection (7)

[More Filters](#)

[Bulk select](#)



No Patent

FIGURE A.7: Ideas: searching for patents and extraction

Patents 36 News Articles 0 Papers 0

Patents News Articles Research Papers Papers (Legacy)

Show only selection (36)

36 Selections < 1 2 3 4 >

Self-balancing scooter and main frame assembly thereof
 US11203390-20211221
 A self-balancing scooter and a main frame assembly thereof are provided. The self-balancing scooter includes a first main frame and a cooperated second main frame. The first main frame includes a first connecting portion, the first connecting portion includes a first end surface, the first end surfa... [Expand](#)
 Selected a minute ago by Guillaume Guarino

Inclination adjustment device for electric scooter steering assembly
 US07210705-20070501
 An adjustment device for the steering assembly of an electric scooter is provided. The adjustment device is at least composed of a body, two confining plates, an elastic element, a cam wheel member, two bearing rods, and a sliding shaft. The sliding shaft has its bottom end pin-joined to a joining sea... [Expand](#)
 Selected a minute ago by Guillaume Guarino

Scooter assemblies and scooter deck assemblies
 US08613457-20131224
 A deck assembly for a scooter comprises a deck having a top surface, a bottom surface, and two side surfaces, and a grind plate having a front end, a rear end, a mating surface, and a grinding surface. The grind plate may be slidably attachable to the deck such that the mating surface of the grind pla... [Expand](#)
 Selected a minute ago by Guillaume Guarino

FIGURE A.8: Ideas: patents selection

3 Build Your Problem Graph

Open Graph

There are 13 patents to extract
 You selected 13 patents that have not been extracted yet. Click below to extract the knowledge from these sources in the form of **problems**, **partial solutions** and **parameters**.

> See details

Extract

FIGURE A.9: Ideas: extraction and problem graph

Appendix B

Neural Networks

Contents

B.1 Formal neuron and neural networks	97
B.2 Training	99
B.2.1 Loss	99
B.2.2 Error back-propagation	100
B.3 Training and specialization	101

Neural networks, which are often referred to as artificial intelligence, are a family of algorithms inspired by the behavior of the human brain and neurons. The purpose is to draw inspiration from the human brain to build algorithms capable of learning complex tasks as quickly as possible. The concept of neurons was introduced in 1943 by McCulloch and Pitts when computer science was still in its infancy. The real integration of this concept in computer science came more than 15 years later with Rosenblatt in 1959. His goal was to understand how the retina works and to recognize patterns in images. Rosenblatt created the perceptron, considered as the first artificial neuron and the basis of neural networks.

The 60's and 70's showed a progressive disinterest in artificial neurons, as perceptrons fail to convince of their interest and research is more oriented towards symbolic approaches. Several major works were carried out in the 80's, notably Hopking who introduced gradient backpropagation in 1982.

Neural networks were massively adopted in the early 2000s with the development of parallel computing capabilities that allowed the construction of more complex and efficient networks.

B.1 Formal neuron and neural networks

The model of the formal neuron (Figure B.1) is based on that of a real neuron with incoming and outgoing links between the neurons (synapses, axons).

The output of a neuron will be the result of the application of an activation function on the linear combination of its inputs plus a constant called bias. The pre-activation z will thus be defined as follows:

$$z = \sum_{i=1}^n x_i w_i + w_0 \quad (\text{B.1})$$

with x_i the input i of the neuron and w_i the weight corresponding to the input i . The weights are in fact parameters associated to the neuron whose value will be learned during the training.

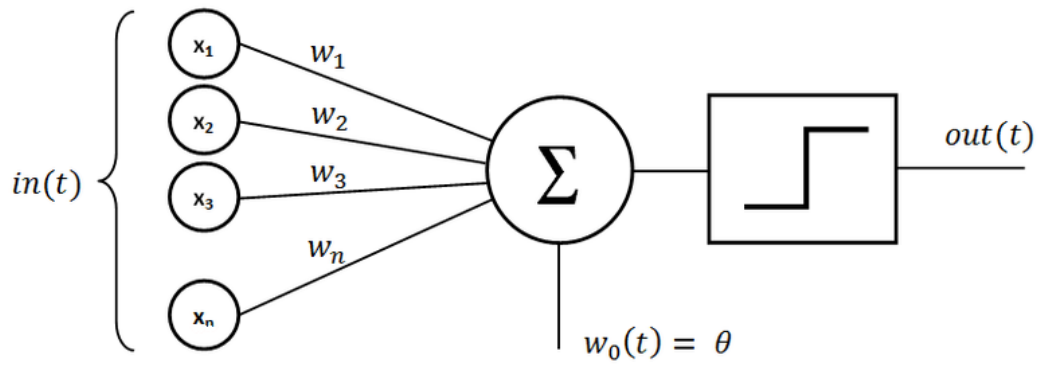


FIGURE B.1: Perceptron

The *sigma* activation function of a perceptron is originally a step function, so the output is binary. The bias replaces the threshold applied to the output to decide when the output should be 1 or 0. However, other activation functions have been developed, the best known of which are the following:

Sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (\text{B.2})$$

Softmax (N is the number of element in x)

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{c=1}^N e^{x_c}} \quad (\text{B.3})$$

Rectified Linear Unit (ReLU)

$$\sigma(x) = \max(0, x) \quad (\text{B.4})$$

TanH

$$\sigma(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (\text{B.5})$$

These activation functions always have the particularity of being non-linear in order to be able to model more complicated behaviors, but they also have the advantage of being differentiable over their entire range of definition, which simplifies the training by gradient descent (Part B.2).

Neural networks are made up of a set of neurons organized in layers as shown in the Figure B.2. Each neuron of a layer will be linked to all the neurons of the previous layer. The neurons of the same layer are not linked. Thus, the outputs of the last layer will be functions of all the parameters of all the neurons and of the inputs of the network. A neural network can thus be modeled by a transfer function f such that:

$$\hat{y} = f(x_i, w_i) \quad (\text{B.6})$$

with \hat{y} the network's output, x_i the inputs and w_i the weights (or parameters) of the network.

The number of neurons in the last layer is related to the task of the network. For a regression (prediction of a real value), one neuron in output will be enough. For classification, there will be as many neurons as there are classes. In this case the i neuron will give the probability of the i class.

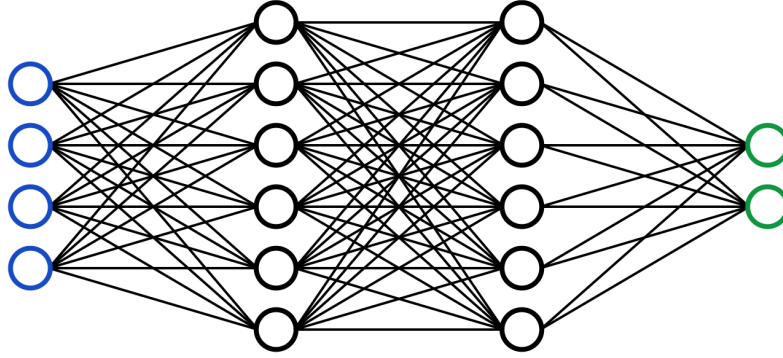


FIGURE B.2: Neural Network

B.2 Training

A neural network can be trained with annotated data i.e. for a set of N observations x^1, x^2, \dots, x^N , the values of the N variables to predict y^1, y^2, \dots, y^N are provided. We define the objective function as the theoretical function f allowing to predict y^i from an observation x^i :

$$\forall i \in [1, N], f(x^i) = y^i \quad (\text{B.7})$$

The goal of training a neural network is to build a surrogate model for this function. The neural network can be seen as a mathematical function g , non-linear and dependent on the values of the parameters associated to each neuron. The training process aims at modifying these parameters so that g approximates f . This type of training is called supervised learning.

B.2.1 Loss

To measure the distance between g and f , a measure called Loss is introduced. The Loss L is computed from the distances between g and f for all known pairs $(x^i, f(x^i) = y^i)$. The most known losses for regression tasks (prediction of real values) are the Mean Absolute Error and Mean Square Error:

Mean Absolute Error (MSE)

$$L = \frac{1}{N} \sum_{i=1}^N |y^i - \hat{y}^i| \quad (\text{B.8})$$

Mean Square Error (MSE)

$$L = \frac{1}{N} \sum_{i=1}^N ||y^i - \hat{y}^i||^2 \quad (\text{B.9})$$

For classification (prediction of the membership to a category called class, e.g. "car", "bus") the cross-entropy and the negative log likelihood are the most commonly used. They are in fact very similar. They both measure how correct the model is on the prediction and the cross entropy has an extra term allowing to take into account the other classes and to measure the weight put in the prediction on the bad classes.

Cross-entropy

$$L = -\frac{1}{N} \sum_{i=1}^N y^i \log(\hat{y}^i) + (1 - y^i) \log(1 - \hat{y}^i) \quad (\text{B.10})$$

Negative log likelihood

$$L = -\frac{1}{N} \sum_{i=1}^N \log(\hat{y}^i) \quad (\text{B.11})$$

B.2.2 Error back-propagation

Once the error between expected and actual predictions is calculated, the network parameters must be updated to decrease this discrepancy. This step is called back-propagation of the error. The name "back-propagation" refers to the mechanism of predicting an output from an input which is called forward propagation.

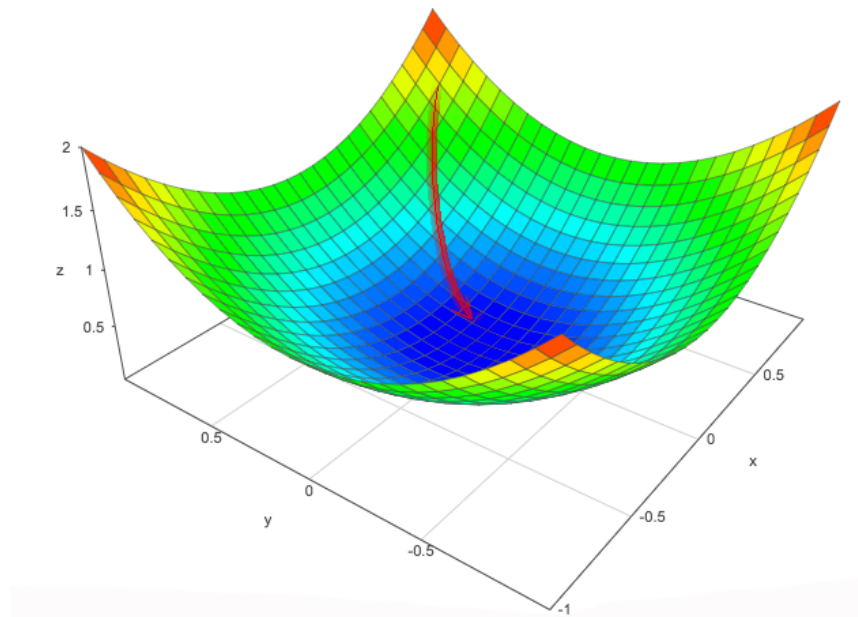


FIGURE B.3: Gradient descent

A simplified representation of the loss as a function of the values of the parameters in the network (here only two parameters) is shown in Figure B.3. At the beginning of the training, the network parameters are randomly initialized and a first loss value is estimated. The goal of the training is to reach the global minimum of the loss function. In a favorable case, with no local minimum, the fastest way is to follow the steepest slope indicated by the gradient $\frac{\partial L}{\partial w}$ (red arrow on Figure B.3). This is the gradient descent. Each parameter is updated in the following way:

$$w = w - lr \frac{1}{N} \sum_{i=1}^N \frac{\delta L_i}{\delta w} \quad (\text{B.12})$$

with lr the learning rate i.e. the speed at which the parameters are updated, and L_i the value of the loss for a couple (x_i, y_i) . The variation of the parameters is then deduced:

$$\Delta w = -lr \frac{\delta L}{\delta w} \quad (\text{B.13})$$

To evaluate the variation of the loss between steps j and $j + 1$, and to verify that it is negative since it is the training purpose, we can use the following formulas:

$$L^{j+1} = L^j + \Delta L \quad (\text{B.14})$$

$$\Delta L = \Delta w \frac{\delta L}{\delta w} \quad (\text{B.15})$$

This amounts to a linear approximation in the neighborhood of the operating point. With the Δw presented previously we obtain:

$$\Delta L = -lr \frac{\delta L}{\delta w} \frac{\delta L}{\delta w} \quad (\text{B.16})$$

and thus:

$$\Delta L = -lr \frac{\delta L^2}{\delta w} \quad (\text{B.17})$$

which is less than or equal to 0 if we choose $lr > 0$. Thus, by applying this strategy, the loss will converge to a minimum. The larger the learning rate lr is, the greater the capacity to exceed the local minimums and the faster the learning will be but the algorithm will also be less stable and will not always converge. Indeed, as everything relies on a linear approximation in the neighborhood of the operating point, the further away from this point with a large lr the more false this approximation is. In most of the current learning strategies, the lr is adjusted during the training with often a high value at the beginning of the training and a lower value at the end.

B.3 Training and specialization

The training of a neural network is supervised, so annotated data (dataset) is required. This data must be numerous to allow optimal learning of each of the network parameters. The smaller the amount of data available, the higher the chances of overfitting. Overfitting occurs when the network performs so well on training data that its ability to generalize and perform well on new data decreases.

However, access to a large amount of annotated data for a specific task is not always guaranteed. This is why transfer learning has been introduced. It consists in training the network on a similar, but different, task. Then, the network is trained on the data associated with the target task, which is less numerous, with a very low learning rate to avoid overfitting.

Other training methods are also possible. Reinforcement learning does not use losses as such but aims at maximizing a "reward" according to the prediction of

the network through trial and error. The gradient is still used but to maximize the reward instead of minimizing the error.

Author publications

This thesis takes place in a strong industrial context, at the intersection of two fields: computer science and innovation. This thesis aims not only to develop intelligent and original applications but also to contribute to the creation of algorithms and models which can adapt to technical contents without loss of performance. The publications of this thesis are described in this chapter. In a first section, publications in computer science related journals and conferences are listed. In a second section, publications in the field of TRIZ and innovation are shown.

Computer Science

Conference proceedings (refereed):

Guarino, Guillaume et al. (2020). “SummaTRIZ : Summarization Networks for Mining Patent Contradiction”. In: 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 979–986. DOI: 10.1109/ICMLA51294.2020.00159
Rank C

Guarino, Guillaume et al. (2021). “PaGAN: Generative Adversarial Network for Patent understanding”. In: 2021 IEEE International Conference on Data Mining (ICDM), pp. 1084–1089. DOI:10.1109/ICDM51629.2021.00126
Rank A*

Guarino, Guillaume et al. (2022). “Réseau antagoniste génératif pour la fouille des contradictions TRIZ dans les brevets”. In: Revue des Nouvelles Technologies de l’Information Extraction et Gestion des Connaissances, RNTI-E-38, pp. 379–386
Rank C

Peer-reviewed journals:

Guarino, Guillaume, Ahmed Samet, and Denis Cavallucci (2022). “PaTRIZ: A framework for mining TRIZ contradictions in patents”. In: Expert Systems with Applications 207, p. 117942. ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2022.117942>. URL: <https://www.sciencedirect.com/science/article/pii/S0957417422011800>.
Quartile Q1

Submitted:

SynCRF: Syntax-based Conditional Random Field for TRIZ parameter mining in *Journal of Intelligent Manufacturing*

TRIZ / Innovation

Conference proceedings (refereed):

Guarino, Guillaume, Ahmed Samet, and Denis Cavallucci (2020). “Summarization as a Denoising Extraction Tool”. In: *Systematic Complex Problem Solving in the Age of Digitalization and Open Innovation*. Ed. by Denis Cavallucci, Stelian Brad, and Pavel Livotov. Cham: Springer International Publishing, pp. 77–87. ISBN: 978-3-030-61295-5

Guarino, Guillaume, Ahmed Samet, and Denis Cavallucci (2021). “Patent Specialization for Deep Learning Information Retrieval Algorithms”. In: *Creative Solutions for a Sustainable Development*. Ed. by Yuri Borgianni et al. Cham: Springer International Publishing, pp. 162–169. ISBN: 978-3-030-86614-3

Guarino, Guillaume and Denis Cavallucci (2022). “Automated TRIZ Domain Mapping”. In: *Systematic Innovation Partnerships with Artificial Intelligence and Information Technology*. Ed. by Robert Nowak, Jerzy Chrzaszcz, and Stelian Brad. Cham: Springer International Publishing, pp. 198–205. ISBN: 978-3-031-17288-5

Bibliography

- Abdelgawad, Louay et al. (2019). "Optimizing neural networks for patent classification". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 688–703.
- Al Saied, Hazem, Nicolas Dugué, and J.-C Lamirel (Sept. 2018). "Automatic summarization of scientific publications using a feature selection approach". In: *International Journal on Digital Libraries* 19, pp. 203–215. DOI: 10.1007/s00799-017-0214-x.
- Allahyari, Mehdi et al. (July 2017). "A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques". In.
- Altshuller, Guenrich (1984). *Creativity As an Exact Science*. CRC Press. ISBN: 9781466593442. URL: <https://books.google.fr/books?id=bUFZDwAAQBAJ>.
- Alvarez-Melis, David and Tommi Jaakkola (2018). "Gromov-Wasserstein Alignment of Word Embedding Spaces". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 1881–1890. DOI: 10.18653/v1/D18-1214. URL: <https://aclanthology.org/D18-1214>.
- Amplayo, Reinald Kim et al. (2018). "Translations as Additional Contexts for Sentence Classification". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. IJCAI'18. Stockholm, Sweden: AAAI Press, 3955–3961. ISBN: 9780999241127.
- Aone, Chinatsu et al. (1997). "A Scalable Summarization System Using Robust NLP". In: *Intelligent Scalable Text Summarization*. URL: <https://www.aclweb.org/anthology/W97-0711>.
- Beltagy, Iz, Matthew E. Peters, and Arman Cohan (2020). "Longformer: The Long-Document Transformer". In: *arXiv:2004.05150*.
- Bender, Oliver, Franz Josef Och, and Hermann Ney (2003). "Maximum Entropy Models for Named Entity Recognition". In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 148–151. URL: <https://aclanthology.org/W03-0420>.
- Bengio, Yoshua et al. (2003). "A Neural Probabilistic Language Model". In: *J. Mach. Learn. Res.* 3, null, 1137–1155. ISSN: 1532-4435.
- Berdyugina, Daria and Denis Cavallucci (Apr. 2022a). "Automatic extraction of inventive information out of patent texts in support of manufacturing design studies using Natural Languages Processing". In: *Journal of Intelligent Manufacturing*, pp. 1–15. DOI: 10.1007/s10845-022-01943-y.
- (Dec. 2022b). "Exploitation of causal relation for automatic extraction of contradiction from a domain-restricted patent corpus". In: *Proceedings of TRIZ Future 2022*. Brussels, Belgium: ETRIA.
- Bojanowski, Piotr et al. (July 2016). "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5. DOI: 10.1162/tacl_a_00051.
- Boufeloussen, Omar and Denis Cavallucci (Sept. 2021). "Bringing Together Engineering Problems and Basic Science Knowledge, One Step Closer to Systematic

- Invention". In: pp. 340–351. ISBN: 978-3-030-86613-6. DOI: 10.1007/978-3-030-86614-3_27.
- Brown, Tom et al. (2020). "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 1877–1901. URL: <https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf>.
- Carreras, Xavier, Lluís Màrquez, and Lluís Padró (Apr. 2003). "Named Entity Recognition For Catalan Using Only Spanish Resources and Unlabelled Data". In: *10th Conference of the European Chapter of the Association for Computational Linguistics*. Budapest, Hungary: Association for Computational Linguistics. URL: <https://aclanthology.org/E03-1038>.
- Cascini, Gaetano and Davide Russo (Jan. 2007). "Computer-aided analysis of patents and search for TRIZ contradictions". In: *International Journal of Product Development* 4, pp. 52–67. DOI: 10.1504/IJPD.2007.011533.
- Cavallucci, Denis (2012). "From TRIZ to Inventive Design Method (IDM): towards a formalization of Inventive Practices in R&D Departments". In.
- Chai, Zhaoying et al. (Jan. 2022). "Hierarchical shared transfer learning for biomedical named entity recognition". In: *BMC Bioinformatics* 23. DOI: 10.1186/s12859-021-04551-4.
- Chang, Hsiang-Tang, Chen-Yen Chang, and Wen-Kuei Wu (May 2017). "Computerized innovation inspired by existing patents". In: pp. 1134–1137. DOI: 10.1109/ICASI.2017.7988268.
- Chen, Dapeng et al. (2018). "Group Consistent Similarity Learning via Deep CRF for Person Re-Identification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Chen, Hui et al. (2019). "GRN: Gated Relation Network to Enhance Convolutional Neural Network for Named Entity Recognition". In: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*. AAAI Press, pp. 6236–6243. DOI: 10.1609/aaai.v33i01.33016236. URL: <https://doi.org/10.1609/aaai.v33i01.33016236>.
- Chiu, Jason P.C. and Eric Nichols (July 2016). "Named Entity Recognition with Bidirectional LSTM-CNNs". In: *Transactions of the Association for Computational Linguistics* 4, pp. 357–370. ISSN: 2307-387X. DOI: 10.1162/tac1_a_00104. eprint: https://direct.mit.edu/tac1/article-pdf/doi/10.1162/tac1_a_00104/1567392/tac1_a_00104.pdf. URL: https://doi.org/10.1162/tac1_a_00104.
- Cho, Kyunghyun et al. (2014). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. ACL, pp. 1724–1734. DOI: 10.3115/v1/d14-1179. URL: <https://doi.org/10.3115/v1/d14-1179>.
- Chu, Xiao et al. (2016). "CRF-CNN: Modeling Structured Information in Human Pose Estimation". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc. URL: <https://proceedings.neurips.cc/paper/2016/file/6f3ef77ac0e3619e98159e9b6febf557-Paper.pdf>.
- Chung, Junyoung et al. (2014). "Empirical evaluation of gated recurrent neural networks on sequence modeling". English (US). In: *NIPS 2014 Workshop on Deep Learning, December 2014*.
- Collobert, Ronan et al. (2010). "Natural Language Processing (almost) from Scratch". In: *Journal of Machine Learning Research*.

- Conroy, John and Dianne O'leary (Jan. 2001). "Text summarization via hidden Markov models". In: *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 406–407. ISBN: 1-58113-331-6. DOI: 10.1145/383952.384042.
- Cremers, Katrin et al. (Aug. 1999). "Citation Frequency And The Value Of Patented Inventions". In: *The Review of Economics and Statistics* 81, pp. 511–515. DOI: 10.1162/003465399558265.
- Croce, Danilo, Giuseppe Castellucci, and Roberto Basili (July 2020). "GAN-BERT: Generative Adversarial Learning for Robust Text Classification with a Bunch of Labeled Examples". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 2114–2119. DOI: 10.18653/v1/2020.acl-main.191. URL: <https://www.aclweb.org/anthology/2020.acl-main.191>.
- Devlin, Jacob et al. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. cite arxiv:1810.04805 Comment: 13 pages. URL: <http://arxiv.org/abs/1810.04805>.
- (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *NAACL-HLT*.
- Dong, Li et al. (2019). "Unified Language Model Pre-training for Natural Language Understanding and Generation". In: *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.
- Dosovitskiy, Alexey, Jost Tobias Springenberg, and Thomas Brox (2015). "Learning to Generate Chairs With Convolutional Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Fix, Evelyn and Joseph L. Hodges (1989). "Discriminatory Analysis - Nonparametric Discrimination: Consistency Properties". In: *International Statistical Review* 57, p. 238.
- Florian, Radu et al. (2003). "Named Entity Recognition through Classifier Combination". In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 168–171. URL: <https://aclanthology.org/W03-0425>.
- Gaschk, Joshua L., Celine H. Frère, and Christofer J. Clemente (Dec. 2019). "Quantifying koala locomotion strategies: implications for the evolution of arborealism in marsupials". In: *Journal of Experimental Biology* 222.24. jeb207506. ISSN: 0022-0949. DOI: 10.1242/jeb.207506. eprint: <https://journals.biologists.com/jeb/article-pdf/222/24/jeb207506/1979966/jeb207506.pdf>. URL: <https://doi.org/10.1242/jeb.207506>.
- Gatt, Albert and Ehud Reiter (Apr. 2009). "SimpleNLG: A realisation engine for practical applications". In: *Proceedings of the 12th European Workshop on Natural Language Generation, ENLG 2009*, pp. 90–93. DOI: 10.3115/1610195.1610208.
- Goodfellow, Ian J. et al. (2014). "Generative Adversarial Nets". In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. NIPS'14*. Montreal, Canada: MIT Press, 2672–2680.
- Gregor, Karol et al. (2015). "DRAW: A Recurrent Neural Network For Image Generation". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 1462–1471. URL: <https://proceedings.mlr.press/v37/gregor15.html>.
- Guarino, Guillaume et al. (2020). "SummaTRIZ : Summarization Networks for Mining Patent Contradiction". In: *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 979–986. DOI: 10.1109/ICMLA51294.2020.00159.

- Guzmán-rivera, Abner, Dhruv Batra, and Pushmeet Kohli (2012). “Multiple Choice Learning: Learning to Produce Multiple Structured Outputs”. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc. URL: <https://proceedings.neurips.cc/paper/2012/file/cfbce4c1d7c425baf21d6b6f2babe6be-Paper.pdf>.
- Habibi, Maryam et al. (July 2017). “Deep learning with word embeddings improves biomedical named entity recognition”. In: *Bioinformatics* 33.14, pp. i37–i48. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx228. eprint: <https://academic.oup.com/bioinformatics/article-pdf/33/14/i37/25157154/btx228.pdf>. URL: <https://doi.org/10.1093/bioinformatics/btx228>.
- Hammerton, James (2003a). “Named Entity Recognition with Long Short-Term Memory”. In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 172–175. URL: <https://aclanthology.org/W03-0426>.
- (2003b). “Named Entity Recognition with Long Short-Term Memory”. In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4. CONLL '03*. Edmonton, Canada: Association for Computational Linguistics, 172–175. DOI: 10.3115/1119176.1119202. URL: <https://doi.org/10.3115/1119176.1119202>.
- Hao, Jia et al. (2019). “An evolutionary computation based method for creative design inspiration generation”. In: *Journal of Intelligent Manufacturing* 30.4, pp. 1673–1691.
- He, Cong and Han Tong Loh (2008). “Grouping of TRIZ Inventive Principles to facilitate automatic patent classification”. In: *Expert Syst. Appl.* 34, pp. 788–795.
- Hermann, Karl Moritz et al. (2015). “Teaching Machines to Read and Comprehend”. In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1. NIPS'15*. Montreal, Canada: MIT Press, 1693–1701.
- Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997a). “Long Short-Term Memory”. In: *Neural Comput.* 9.8, 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- (1997b). “Long Short-Term Memory”. In: *Neural Comput.* 9.8, 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Irie, Kazuki et al. (2019). “Language Modeling with Deep Transformers”. In: *Proc. Interspeech 2019*, pp. 3905–3909. DOI: 10.21437/Interspeech.2019-2225.
- Jardim-Goncalves, Ricardo et al. (2011). “Knowledge framework for intelligent manufacturing systems”. In: *Journal of Intelligent Manufacturing* 22.5, pp. 725–735.
- Jugulum, Rajesh and Daniel D. Frey (2007). “Toward a taxonomy of concept designs for improved robustness”. In: *Journal of Engineering Design* 18.2, pp. 139–156. DOI: 10.1080/09544820600731496.
- Kim, Sunhye, Incha Park, and Byungun Yoon (Feb. 2020). “SAO2Vec: Development of an algorithm for embedding the subject–action–object (SAO) structure using Doc2Vec”. In: *PLOS ONE* 15.2, pp. 1–26. DOI: 10.1371/journal.pone.0227930. URL: <https://doi.org/10.1371/journal.pone.0227930>.
- Kipf, Thomas N. and Max Welling (2017). “Semi-Supervised Classification with Graph Convolutional Networks”. In: *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. ICLR '17. Palais des Congrès Neptune, Toulon, France. URL: <https://openreview.net/forum?id=SJU4ayYgl>.
- Klein, Dan et al. (2003). “Named Entity Recognition with Character-Level Models”. In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pp. 180–183. URL: <https://aclanthology.org/W03-0428>.

- Kleinberg, Jon (Jan. 1999). "Authoritative Sources in a Hyperlinked Environment". In: *Journal of The ACM - JACM* 46.
- Konishi, Kazuya (2005). "Query Terms Extraction from Patent Document for Invalidity Search". In: *NTCIR*.
- Korobkin, D. M., S. A. Fomenkov, and A. G. Kravets (2017). "Extraction of physical effects practical applications from patent database". In: *2017 8th International Conference on Information, Intelligence, Systems Applications (IISA)*, pp. 1–5. DOI: 10.1109/IISA.2017.8316402.
- Kupiec, J., J. Pedersen, and F. Chen (1995). "A trainable document summarizer". In: *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Seattle, Washington, United States: ACM Press, pp. 68–73. ISBN: 0-89791-714-6.
- Kusar, Janez et al. (2004). "How to reduce new product development time". In: *Robotics and Computer-Integrated Manufacturing* 20.1, pp. 1–15. ISSN: 0736-5845. DOI: [https://doi.org/10.1016/S0736-5845\(03\)00049-8](https://doi.org/10.1016/S0736-5845(03)00049-8). URL: <https://www.sciencedirect.com/science/article/pii/S0736584503000498>.
- Kusiak, Andrew (2016). "Put innovation science at the heart of discovery". In: *Nature* 530.7590, pp. 255–255.
- Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira (2001). "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 282–289. ISBN: 1-55860-778-1. URL: <http://dl.acm.org/citation.cfm?id=645530.655813>.
- Lamirel, Jean-Charles et al. (2004). "New classification quality estimators for analysis of documentary information: application to patent analysis and web mapping". In: *Scientometrics* 60.3. Article dans revue scientifique avec comité de lecture. internationale., pp. 445–462. URL: <https://hal.inria.fr/inria-00100074>.
- Lample, Guillaume et al. (June 2016). "Neural Architectures for Named Entity Recognition". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 260–270. DOI: 10.18653/v1/N16-1030. URL: <https://aclanthology.org/N16-1030>.
- Lan, Zhenzhong et al. (2019). *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. arXiv: 1909.11942 [cs.CL].
- Lanjouw, Jean and Mark Schankerman (Feb. 2004). "Patent Quality and Research Productivity: Measuring Innovation with Multiple Indicators". In: *Economic Journal* 114, pp. 441–465. DOI: 10.1111/j.1468-0297.2004.00216.x.
- Larkey, Leah S. (1997). "Some Issues in the Automatic Classification of US Patents". In.
- Le, Quoc and Tomas Mikolov (2014). "Distributed Representations of Sentences and Documents". In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. ICML'14. Beijing, China: JMLR.org, II-1188–II-1196.
- Lecun, Y. et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324. DOI: 10.1109/5.726791.
- Lee, Jieh-Sheng and Jieh Hsiang (2020a). "Patent claim generation by fine-tuning OpenAI GPT-2". In: *World Patent Information* 62, p. 101983. ISSN: 0172-2190. DOI: <https://doi.org/10.1016/j.wpi.2020.101983>. URL: <https://www.sciencedirect.com/science/article/pii/S0172219019300766>.

- Lee, Jieh-Sheng and Jieh Hsiang (2020b). "Patent classification by fine-tuning BERT language model". In: *World Patent Information* 61, p. 101965. ISSN: 0172-2190. DOI: <https://doi.org/10.1016/j.wpi.2020.101965>. URL: <https://www.sciencedirect.com/science/article/pii/S0172219019300742>.
- Lewis, Mike et al. (July 2020a). "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703. URL: <https://aclanthology.org/2020.acl-main.703>.
- (2020b). "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. DOI: 10.18653/v1/2020.acl-main.703. URL: <https://doi.org/10.18653/v1/2020.acl-main.703>.
- Li, Pengfei et al. (2021). "ACT: an Attentive Convolutional Transformer for Efficient Text Classification". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.15, pp. 13261–13269. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/17566>.
- Li, Xiangyang, Huan Zhang, and Xiao-Hua Zhou (2020). "Chinese clinical named entity recognition with variant neural structures based on BERT methods". In: *Journal of Biomedical Informatics* 107, p. 103422. ISSN: 1532-0464. DOI: <https://doi.org/10.1016/j.jbi.2020.103422>. URL: <https://www.sciencedirect.com/science/article/pii/S1532046420300502>.
- Li, Zhen et al. (Oct. 2012). "A Framework for Automatic TRIZ Level of Invention Estimation of Patents Using Natural Language Processing, Knowledge-Transfer and Patent Citation Metrics". In: *Comput. Aided Des.* 44.10, 987–1010. ISSN: 0010-4485. DOI: 10.1016/j.cad.2011.12.006. URL: <https://doi.org/10.1016/j.cad.2011.12.006>.
- Liang, Y. et al. (2009). "Computer-aided classification of patents oriented to TRIZ". In: *2009 IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 2389–2393.
- Liang, Yanhong and Runhua Tan (2007). "A Text-Mining-based Patent Analysis in Product Innovative Process". In: *Trends in Computer Aided Innovation*. Ed. by Noel León-Rovira. Boston, MA: Springer US, pp. 89–96. ISBN: 978-0-387-75456-7.
- Litvak, Marina and Mark Last (Aug. 2008). "Graph-based keyword extraction for single-document summarization". In: pp. 17–24. DOI: 10.3115/1613172.1613178.
- Liu, Linqing et al. (Nov. 2017). "Generative Adversarial Network for Abstractive Text Summarization". In.
- Liu, Yang and Mirella Lapata (Nov. 2019). "Text Summarization with Pretrained Encoders". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 3730–3740. DOI: 10.18653/v1/D19-1387. URL: <https://aclanthology.org/D19-1387>.
- Loh, Han, Cong He, and Lixiang Shen (Mar. 2006). "Automatic classification of patent documents for TRIZ users". In: *World Patent Information* 28, pp. 6–13. DOI: 10.1016/j.wpi.2005.07.007.
- Lu, Yonghe et al. (2020). "Research on classification and similarity of patent citation based on deep learning". In: *Scientometrics* 123.2, pp. 813–839. DOI: 10.1007/s11192-020-03385-. URL: https://ideas.repec.org/a/spr/scient/v123y2020i2d10.1007_s11192-020-03385-w.html.

- Ma, Xuezhe and Eduard Hovy (Aug. 2016). "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1064–1074. DOI: 10.18653/v1/P16-1101. URL: <https://aclanthology.org/P16-1101>.
- Mahdabi, Parvaz et al. (2011). "Building Queries for Prior-Art Search". In: *Multidisciplinary Information Retrieval*. Ed. by Allan Hanbury, Andreas Rauber, and Arjen P. de Vries. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 3–15. ISBN: 978-3-642-21353-3.
- Mahdabi, Parvaz et al. (2012). "Automatic Refinement of Patent Queries Using Concept Importance Predictors". In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '12. Portland, Oregon, USA: Association for Computing Machinery, 505–514. ISBN: 9781450314725. DOI: 10.1145/2348283.2348353. URL: <https://doi.org/10.1145/2348283.2348353>.
- Melis, Gábor, Tomáš Kočiský, and Phil Blunsom (2020). "Mogriifier LSTM". In: *International Conference on Learning Representations*. URL: <https://openreview.net/forum?id=SJe5P6EYvS>.
- Mihalcea, Rada (2004). "Graph-based Ranking Algorithms for Sentence Extraction Applied to Text Summarization". In: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, companion volume (ACL 2004)*. URL: <http://www.cs.unt.edu/~rada/papers.html>.
- Mikolov, Tomas et al. (2013a). "Distributed Representations of Words and Phrases and Their Compositionality". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'13. Lake Tahoe, Nevada: Curran Associates Inc., pp. 3111–3119. URL: <http://dl.acm.org/citation.cfm?id=2999792.2999959>.
- Mikolov, Tomas et al. (Jan. 2013b). "Efficient Estimation of Word Representations in Vector Space". In: *Efficient Estimation of Word Representations in Vector Space*, pp. 1–12.
- Morin, Frederic and Yoshua Bengio (2005). "Hierarchical probabilistic neural network language model". In: *AISTATS'05*, pp. 246–252.
- Mou, Lili et al. (2015). "Discriminative Neural Sentence Modeling by Tree-Based Convolution". In: *EMNLP*.
- Nallapati, Ramesh, Feifei Zhai, and Bowen Zhou (2017). "SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI'17. San Francisco, California, USA: AAAI Press, 3075–3081.
- Nallapati, Ramesh et al. (Aug. 2016). "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond". In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, pp. 280–290. DOI: 10.18653/v1/K16-1028. URL: <https://www.aclweb.org/anthology/K16-1028>.
- Narayan, Shashi, Shay B. Cohen, and Mirella Lapata (2018a). "Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 1797–1807. DOI: 10.18653/v1/D18-1206. URL: <https://www.aclweb.org/anthology/D18-1206>.

- Narayan, Shashi, Shay B. Cohen, and Mirella Lapata (June 2018b). "Ranking Sentences for Extractive Summarization with Reinforcement Learning". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 1747–1759. URL: <https://www.aclweb.org/anthology/N18-1158>.
- Ni, Xin, Ahmed Samet, and Denis Cavallucci (2022). "Similarity-based approach for inventive design solutions assistance". In: *J. Intell. Manuf.* 33, pp. 1681–1698.
- Ni, Xin et al. (2021). "PatRIS: Patent Ranking Inventive Solutions". In: *Database and Expert Systems Applications*. Ed. by Christine Strauss et al. Cham: Springer International Publishing, pp. 295–309. ISBN: 978-3-030-86475-0.
- Oya, Tatsuro et al. (June 2014). "A Template-based Abstractive Meeting Summarization: Leveraging Summary and Source Text Relationships". In: *Proceedings of the 8th International Natural Language Generation Conference (INLG)*. Philadelphia, Pennsylvania, U.S.A.: Association for Computational Linguistics, pp. 45–53. DOI: 10.3115/v1/W14-4407. URL: <https://aclanthology.org/W14-4407>.
- Page, Larry et al. (1998). *The PageRank Citation Ranking: Bringing Order to the Web*.
- Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio (2013). "On the difficulty of training recurrent neural networks". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, pp. 1310–1318. URL: <https://proceedings.mlr.press/v28/pascanu13.html>.
- Peng, Jian, Liefeng Bo, and Jinbo Xu (2009). "Conditional Neural Fields". In: *Advances in Neural Information Processing Systems*. Ed. by Y. Bengio et al. Vol. 22. Curran Associates, Inc. URL: <https://proceedings.neurips.cc/paper/2009/file/e820a45f1dfc7b95282d10b6087e11c0-Paper.pdf>.
- Pennington, Jeffrey, Richard Socher, and Christopher Manning (Jan. 2014). "Glove: Global Vectors for Word Representation". In: *EMNLP*. Vol. 14, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.
- Peters, Matthew E. et al. (June 2018). "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: <https://aclanthology.org/N18-1202>.
- Prasad, Biren, Fujun Wang, and Jiati Deng (1998). "A concurrent workflow management process for integrated product development". In: *Journal of Engineering Design* 9.2, pp. 121–135.
- Prasad, Brian (Dec. 1996a). *Concurrent Engineering Fundamentals, Vol I: Integrated Product and Process Organization*. Vol. 1. ISBN: ISBN-10: 0131474634 • ISBN-13: 9780131474635. DOI: 10.13140/RG.2.1.2613.0005.
- (Sept. 1996b). *Concurrent Engineering Fundamentals, Volume II: Integrated Product Development*. Vol. 2, p. 528. ISBN: ISBN-10: 0133969460; ISBN-13: 978-0133969467. DOI: 10.13140/RG.2.1.4710.1527.
- Radford, A. et al. (2019). "Language Models are Unsupervised Multitask Learners". In: *arXiv preprint arXiv:1910.10683* [cs.LG].
- Raffel, Colin et al. (2019). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. arXiv: 1910.10683 [cs.LG].

- Renjith, Sarath C, Kijung Park, and Gül E Okudan Kremer (2020). "A design framework for additive manufacturing: Integration of additive manufacturing capabilities in the early design process". In: *International Journal of Precision Engineering and Manufacturing* 21.2, pp. 329–345.
- Risch, Julian, Samuele Garda, and Ralf Krestel (2020). "Hierarchical Document Classification as a Sequence Generation Task". In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*. New York, NY, USA: Association for Computing Machinery, 147–155. ISBN: 9781450375856. URL: <https://doi.org/10.1145/3383583.3398538>.
- Saha, Tulika, Sriparna Saha, and Pushpak Bhattacharyya (2018). "Exploring Deep Learning Architectures Coupled with CRF Based Prediction for Slot-Filling". In: *Neural Information Processing*. Ed. by Long Cheng, Andrew Chi Sing Leung, and Seiichi Ozawa. Cham: Springer International Publishing, pp. 214–225. ISBN: 978-3-030-04167-0.
- Salimans, Tim et al. (2016). "Improved Techniques for Training GANs". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc. URL: <https://proceedings.neurips.cc/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf>.
- Savransky, Semyon (2000). *Engineering of Creativity: Introduction to TRIZ Methodology of Inventive Problem Solving*. CRC Press. ISBN: 9781420038958. URL: <https://books.google.fr/books?id=a5LMBQAAQBAJ>.
- See, Abigail, Peter J. Liu, and Christopher D. Manning (2017). "Get To The Point: Summarization with Pointer-Generator Networks". In: *ACL*.
- Sennrich, Rico, Barry Haddow, and Alexandra Birch (Aug. 2016). "Neural Machine Translation of Rare Words with Subword Units". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725. DOI: 10.18653/v1/P16-1162. URL: <https://www.aclweb.org/anthology/P16-1162>.
- Shalaby, Marawan et al. (n.d.). "An LSTM Approach to Patent Classification based on Fixed Hierarchy Vectors". In: *Proceedings of the 2018 SIAM International Conference on Data Mining (SDM)*, pp. 495–503. DOI: 10.1137/1.9781611975321.56. eprint: <https://epubs.siam.org/doi/pdf/10.1137/1.9781611975321.56>. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9781611975321.56>.
- Shen, Dou et al. (Jan. 2007). "Document Summarization Using Conditional Random Fields." In: *IJCAI International Joint Conference on Artificial Intelligence*, pp. 2862–2867.
- Shinmori, Akihiro and Yuzo Marukawa (Jan. 2003). "Patent claim processing for readability: Structure analysis and term explanation". In: DOI: 10.3115/1119303.1119310.
- Shirwaiker, Rohan A and Gül E Okudan (2008). "Triz and axiomatic design: a review of case-studies and a proposed synergistic use". In: *Journal of Intelligent Manufacturing* 19.1, pp. 33–47.
- Smirnov, Alexander et al. (2013). "Knowledge management for complex product development". In: *IFIP International Conference on Product Lifecycle Management*. Springer, pp. 110–119.
- Smith, Shaden et al. (2022). "Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model". In: *CoRR abs/2201.11990*. arXiv: 2201.11990. URL: <https://arxiv.org/abs/2201.11990>.
- Song, Kaitao et al. (2019). "MASS: Masked Sequence to Sequence Pre-training for Language Generation". In: *CoRR abs/1905.02450*. arXiv: 1905.02450. URL: <http://arxiv.org/abs/1905.02450>.

- Song, Kaitao et al. (2020). "MPNet: Masked and Permuted Pre-training for Language Understanding". In: *NeurIPS 2020*. ACM. URL: <https://www.microsoft.com/en-us/research/publication/mpnet-masked-and-permuted-pre-training-for-language-understanding/>.
- Souili, Achille and Denis Cavallucci (July 2017). "Automated Extraction of Knowledge Useful to Populate Inventive Design Ontology from Patents". In: pp. 43–62. ISBN: 978-3-319-56592-7. DOI: 10.1007/978-3-319-56593-4_2.
- Sperduti, A. and A. Starita (1997). "Supervised neural networks for the classification of structures". In: *IEEE Transactions on Neural Networks* 8.3, pp. 714–735. DOI: 10.1109/72.572108.
- Sun, Junlin et al. (Mar. 2022). "Deep learning-based methods for natural hazard named entity recognition". In: *Scientific Reports* 12, p. 4598. DOI: 10.1038/s41598-022-08667-2.
- Vaswani, Ashish et al. (2017). "Attention is All you Need". In: *Advances in Neural Information Processing Systems* 30. Ed. by I. Guyon et al. Curran Associates, Inc., pp. 5998–6008. URL: <http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>.
- Vemulapalli, Raviteja et al. (2016). "Gaussian Conditional Random Field Network for Semantic Segmentation". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3224–3233. DOI: 10.1109/CVPR.2016.351.
- Wang, Gangfeng et al. (Mar. 2016). "Extraction of Principle Knowledge from Process Patents for Manufacturing Process Innovation". In: vol. 56, pp. 193–198. DOI: 10.1016/j.procir.2016.10.053.
- Wang, Qicai et al. (2019). "A Text Abstraction Summary Model Based on BERT Word Embedding and Reinforcement Learning". In: *Applied Sciences* 9.21. ISSN: 2076-3417. DOI: 10.3390/app9214701. URL: <https://www.mdpi.com/2076-3417/9/21/4701>.
- Weston, Jason, Frédéric Ratle, and Ronan Collobert (2008). "Deep Learning via Semi-Supervised Embedding". In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland: Association for Computing Machinery, 1168–1175. ISBN: 9781605582054. DOI: 10.1145/1390156.1390303. URL: <https://doi.org/10.1145/1390156.1390303>.
- Widdowson, Henry (Oct. 2007). "J.R. Firth, 1957, Papers in Linguistics 1934–51". In: *International Journal of Applied Linguistics* 17, pp. 402–413. DOI: 10.1111/j.1473-4192.2007.00164.x.
- Wu, Yonghui et al. (Sept. 2016). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". In.
- Xu, H. et al. (2018). "Sequence Generative Adversarial Network for Long Text Summarization". In: *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 242–248. DOI: 10.1109/ICTAI.2018.00045.
- Xu, Lei et al. (2021). "Named Entity Recognition of BERT-BiLSTM-CRF Combined with Self-attention". In: *Web Information Systems and Applications*. Ed. by Chunxiao Xing et al. Cham: Springer International Publishing, pp. 556–564. ISBN: 978-3-030-87571-8.
- Xue, Xiaobing and W. Bruce Croft (2009). "Transforming Patents into Prior-Art Queries". In: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '09. Boston, MA, USA: Association for Computing Machinery, 808–809. ISBN: 9781605584836. DOI: 10.1145/1571941.1572139. URL: <https://doi.org/10.1145/1571941.1572139>.

- Yang, Zhilin, William W. Cohen, and Ruslan Salakhutdinov (2016). "Revisiting Semi-Supervised Learning with Graph Embeddings". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, 40–48.
- Yang, Zhilin et al. (2019a). "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., pp. 5754–5764. URL: <http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.pdf>.
- Yang, Zhilin et al. (2019b). "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc.
- Yanhong Liang, Runhua Tan, and Jianhong Ma (2008). "Patent analysis with text mining for TRIZ". In: *2008 4th IEEE International Conference on Management of Innovation and Technology*, pp. 1147–1151.
- Yu, Xin et al. (2019). "BioBERT Based Named Entity Recognition in Electronic Medical Record". In: *2019 10th International Conference on Information Technology in Medicine and Education (ITME)*, pp. 49–52. DOI: 10.1109/ITME.2019.00022.
- Yuan, Zehuan, Tong Lu, and Yirui Wu (2017). "Deep-Dense Conditional Random Fields for Object Co-Segmentation". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence. IJCAI'17*. Melbourne, Australia: AAAI Press, 3371–3377. ISBN: 9780999241103.
- Zhang, Jingqing et al. (2019). "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization". In: *ArXiv abs/1912.08777*.
- Zheng, Shuai et al. (2015). "Conditional Random Fields as Recurrent Neural Networks". In: *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1529–1537. DOI: 10.1109/ICCV.2015.179.
- Zhong, Ming et al. (Apr. 2020). "Extractive Summarization as Text Matching". In: *Proceedings of the 2018 Conference of the North*.
- Zhou, Peng et al. (2016). "Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling." In: *COLING*. Ed. by Nicoletta Calzolari, Yuji Matsumoto, and Rashmi Prasad. ACL, pp. 3485–3495. ISBN: 978-4-87974-702-0. URL: <http://dblp.uni-trier.de/db/conf/coling/coling2016.html#ZhouQZXB16>.
- Zhou, Qingyu et al. (July 2018). "Neural Document Summarization by Jointly Learning to Score and Select Sentences". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 654–663. DOI: 10.18653/v1/P18-1061. URL: <https://www.aclweb.org/anthology/P18-1061>.
- Zhu, Huiming et al. (Jan. 2020). "Patent Automatic Classification Based on Symmetric Hierarchical Convolution Neural Network". In: *Symmetry* 12, p. 186. DOI: 10.3390/sym12020186.
- Zhuang, Liu et al. (Aug. 2021). "A Robustly Optimized BERT Pre-training Approach with Post-training". English. In: *Proceedings of the 20th Chinese National Conference on Computational Linguistics*. Huhhot, China: Chinese Information Processing Society of China, pp. 1218–1227. URL: <https://aclanthology.org/2021.ccl-1.108>.

UNIVERSITÉ DE STRASBOURG

THÈSE DE DOCTORAT : RÉSUMÉ

**Fouille de textes pour l'automatisation du
processus de conception inventive inspiré
de TRIZ par l'usage des brevets**

Auteur :
Guillaume GUARINO

Une thèse soumise pour l'obtention du diplôme de
Docteur
en
Informatique

Thèse soutenue publiquement le 09/12/2022 devant un jury composé de :

Directeur de thèse :	Denis Cavallucci	Professeur, INSA Strasbourg
Encadrant :	Ahmed Samet	Maître de Conférence, INSA Strasbourg
Rapporteuse :	Gül E. Kremer	Professeure, University of Dayton
Rapporteur :	Jean-Charles Lamirel	Maître de Conférence (HDR), Université de Strasbourg
Examinatrice :	Sihem Amer-Yahia	Directeur de Recherche CNRS, Université Grenoble Alpes
Examineur :	Pierre Gañarski	Professeur, Université de Strasbourg
Examineur :	Alexandre Termier	Professeur, Université de Rennes 1

Introduction

Le degré d'innovation d'un produit est l'un des indicateurs clés qui influencent la façon dont ce produit est positionné parmi ses concurrents. Ce degré d'innovation peut être mesuré par la performance du produit dans sa fonction primaire mais pas seulement (par exemple, l'impact environnemental, un coût inférieur pour une performance comparable, etc.). Le point de départ d'un cycle de développement est un problème souvent rencontré par les démarches de conception de produits. Or, au début d'un processus de développement, il n'y a actuellement aucune certitude quant aux performances globales du produit résultant. La théorie de la résolution inventive des problèmes (TRIZ) a été proposée pour favoriser l'innovation de rupture et maximiser les chances de résoudre un problème de manière inventive. Altshuller, le fondateur de cette théorie, présente un pseudo-algorithme allant de la formulation du problème à sa résolution dans le but d'aider les ingénieurs. Sa théorie repose sur un constat fort : chaque problème d'un domaine A peut être relié à un problème déjà résolu dans un domaine B, dont il suffirait d'adapter la solution.

Cependant, trouver un problème correspondant dans un autre domaine nécessite soit de lire toutes les publications scientifiques et brevets existants, soit de développer un algorithme capable de caractériser les problèmes résolus par chaque solution existante. Nous avons choisi de développer nos recherches autour de la seconde option.

Afin de permettre une recherche de solutions inter-domaines, des approches basées sur les réseaux de neurones sont développées pour identifier les concepts clés de la théorie TRIZ dans les brevets. En particulier, nous proposons une approche de fouille de contradictions basée sur le résumé automatique, une approche de fouille de paramètres intégrant la structure syntaxique des phrases et un modèle multi-tâches unifié montrant des performances inégalées. Des jeux de données correspondant aux différentes tâches sont également développés. Des démonstrateurs logiciels sont construits pour montrer la viabilité et l'efficacité de notre approche.

Identification des contradictions TRIZ

Une contradiction TRIZ correspond à une relation particulière entre deux phrases. La première phrase présente un paramètre d'un système, qui, lorsqu'il est amélioré, amène la dégradation d'un paramètre situé dans la deuxième phrase. Les approches de résumé qui cherchent à construire des représentations riches des phrases sont donc particulièrement adaptées pour l'extraction de contradictions. Un jeu de données de contradiction TRIZ a été créé. Il est constitué de 1600 brevets américains dont sont extraites les phrases contenant les paramètres des contradictions.

Le modèle de base, appelé SummaTRIZ, est basé sur un modèle de résumé extractif présenté par LIU et LAPATA, 2019. L'encodeur BERT (DEVLIN et al., 2019) prend une série de tokens en entrée. Chaque phrase est séparée par un token spécial [SEP]. Un autre token spécial [CLS] est utilisé pour représenter chaque phrase. BERT construit une représentation contextuelle pour chaque token. Ces représentations intègrent un maximum d'informations provenant de tokens adjacents. Une couche de Transformer au-dessus de BERT et qui ne prend en entrée que les représentations des tokens [CLS], qui sont donc les représentations des phrases, permet d'avoir une attention globale sur toute la séquence même si celle-ci était plus longue que la limite de 512 tokens pour BERT. La limite de longueur est fixée à 1500 tokens pour s'adapter à la longueur des parties de l'état de l'art du brevet. Le but de cette couche

est aussi de modéliser les relations entre les phrases avec le même processus que les relations entre les tokens dans les couches précédentes.

Le modèle SummaTRIZ est pré-entraîné sur une tâche de résumé extractif d'articles de presse avec le jeu de données CNN/DailyMail HERMANN et al., 2015. L'objectif est d'entraîner la couche d'attention au-dessus de BERT afin qu'elle soit capable de construire une représentation contenant les informations essentielles dans les séquences d'entrée. En effet, l'entraînement de cette couche nécessite une grande variété de documents.

Deux classifieurs (ANN avec deux classes) sont utilisés, au-dessus de la couche supplémentaire de Transformer, pour prédire si chaque phrase appartient à la première partie de la contradiction, à la deuxième partie de la contradiction ou à aucune d'entre elles. Deux classifieurs différents sont utilisés car il existe une probabilité non nulle qu'une phrase contienne la totalité de la contradiction et doit donc être classée à la fois comme première et deuxième partie de la contradiction.

L'apprentissage semi-supervisé vise à améliorer la généralisation d'un modèle à partir de données non étiquetées. Il améliore également la qualité des représentations générées par le modèle WESTON, RATLE et COLLOBERT, 2008; YANG, COHEN et SALAKHUTDINOV, 2016; KIPF et WELLING, 2017. Les réseaux adversariaux génératifs ont été introduits par Goodfellow et al. GOODFELLOW et al., 2014. L'objectif des GANs est de générer de nouvelles données proches d'une distribution de données cible. Les GANs peuvent être adaptés à l'apprentissage semi-supervisé SALIMANS et al., 2016. CROCE, CASTELLUCCI et BASILI, 2020 a montré l'efficacité de cette méthode pour la classification de phrases.

Un modèle appelé générateur G génère des données factices et un autre modèle appelé discriminant D tente de distinguer les données générées parmi des exemples réels mais non étiquetés.

Le générateur G apprend à mettre en correspondance les variables latentes en entrée z avec la distribution des données réelles p_{data} . Son objectif est donc de minimiser $\log(1 - D(G(z)))$. Le discriminant, au contraire, essaie de maximiser $\log(1 - D(G(z)))$ tout en associant les bonnes étiquettes aux données réelles x , c'est-à-dire en maximisant $\log(D(x))$.

Ainsi, D et G jouent un jeu minimax à deux joueurs avec une fonction de valeur $V(G, D)$:

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]. \quad (1)$$

Un entraînement antagoniste pour les classifieurs de phrases est mis en œuvre comme décrit par Salimans et al. SALIMANS et al., 2016. Une classe supplémentaire est créée pour introduire les probabilités d'ajustement des données d'entrée à la distribution cible p_{data} . Les classifieurs de phrases ont donc trois neurones en sortie : les neurones de classification de la contradiction initiale et un autre neurone qui indique la probabilité que le document soit généré. Les neurones de "contradiction" sont donc impliqués dans la perte supervisée (D_{sup_1} ou D_{sup_2}) pour les deux classes de classification de la contradiction C et \bar{C} (la phrase appartient à la partie i de la contradiction ou non, Equation 2) tandis que le neurone "antagoniste" est impliqué dans les pertes non supervisées pour les classes *généré* G et *réel* \bar{G} (Equations 3 et 4).

$$D_{sup_i} = E_{x, y \sim p_{data}}[-\log(P(\hat{y}_{s_i} = y_{s_i} | x, y_{s_i} \in (C, \bar{C})))] \quad (2)$$

$$D_{non-sup_i} = E_{x \sim p_{data}}[-\log(P(\hat{y}_{s_i} = y_{s_i} | x, y_{s_i} = \bar{G}))] \quad (3)$$

$$D_{gen_i} = E_{z \sim p_z} [-\log(P(\hat{y}_{s_i} = y_{s_i} | x, y_{s_i} = G))] \quad (4)$$

$D_{non-sup}$ indique la capacité du modèle à classer les données réelles comme des données réelles. D_{gen} concerne la capacité du modèle à classer les données générées comme des données générées. Les phrases utilisées pour les pertes non supervisées proviennent de brevets non étiquetés. $D_{non-sup}$ est donc rétro-propagé à travers la couche de Transformer et l'encodeur. Cela implique que l'encodeur apprend à intégrer de nouvelles informations dans les représentations des phrases pour permettre une meilleure reconnaissance des données "réelles". Cette représentation plus riche des phrases des brevets induit une classification plus facile pour la contradiction qui est notre objectif principal. Ce mécanisme permet également de minimiser l'overfitting. Ceci est encore amplifié par le fait que deux pertes non-supervisées sont calculées et rétro-propagées en même temps (une pour chaque classifieur).

L'apprentissage antagoniste permet une amélioration des résultats de l'ordre de 20%. Environ un tiers des contradictions sont correctement extraites mais le nombre de faux positifs reste aux alentours de 50%. Cela s'explique par la nature extrêmement complexe de la tâche.

Extraction des paramètres

Une contradiction est une relation particulière entre deux paramètres. Si les phrases décrivant de telles relations sont identifiées, ces paramètres doivent encore être extraits. Les paramètres sont constitués d'un ou plusieurs mots. L'extraction des paramètres est donc similaire, dans sa forme, à une tâche de reconnaissance d'entités nommées (NER).

Dans le cas des paramètres TRIZ, nous remarquons que les structures syntaxiques qui les introduisent sont souvent similaires mais pas identiques. Ceci est principalement dû aux verbes d'évolution (augmenter, diminuer, améliorer...) qui se trouvent souvent à proximité des paramètres. Un encodeur avec un classifieur softmax va prédire une étiquette pour chaque token sans prendre en compte les étiquettes des tokens voisins. Cependant, dans l'exemple précédent, nous pouvons voir que si *Paper* fait partie d'un *EP*, *quality* est susceptible d'en faire partie également. Il semble donc légitime d'essayer d'introduire une dépendance entre les étiquettes prédites avec un champ aléatoire conditionnel (CRF).

Nous nous sommes donc attelés à l'extraction des paramètres TRIZ avec une approche basée sur les CRFs. Un nouveau CRF est introduit, appelé SynCRF, afin de prendre en compte la spécificité syntaxique des phrases de brevet et d'obtenir des résultats optimaux dans l'exploration des paramètres.

L'utilisation d'un CRF au-dessus d'un encodeur permet de tirer parti des représentations contextuelles des tokens à la sortie de l'encodeur. Les modèles de langage masqué, du fait de leur entraînement, intègrent de riches informations syntaxiques. Il est donc intéressant d'étudier la génération des potentiels par paire du champ aléatoire conditionnel à partir de ces représentations contextuelles plutôt que d'utiliser un processus de reconnaissance des parties du discours. De plus, le processus de reconnaissance des parties du discours ajoute de la complexité au calcul. Un réseau de neurones calcule les potentiels à partir des représentations des tokens. Trois configurations différentes sont mises en œuvre pour ce réseau neuronal. Des réseaux neuronaux entièrement connectés à une et deux couches sont testés, ainsi qu'un réseau neuronal récurrent. L'objectif de cette dernière configuration est de construire

un lien direct entre les potentiels par paire générés afin d'améliorer la cohérence des séquences d'étiquettes.

Les résultats montrent une amélioration comprise entre 5 et 10% selon la nature des paramètres par rapport à un modèle réseau de neurones + CRF classique.

Modèle multi-tâches

Le développement de deux modèles différents pour chacune des tâches (contradiction et paramètres) pose la question de leur intégration / fusion. Deux modèles, l'un modulaire appelé modèle naïf, l'autre multi-tâche ont donc été développés.

Le modèle naïf permet l'extraction d'une contradiction (c'est-à-dire deux paramètres, un paramètre par partie de la contradiction) à partir d'un brevet dans un processus en trois étapes. Une phrase est d'abord sélectionnée pour chacune des deux parties de la contradiction en utilisant le modèle de résumé. Les paramètres de ces deux phrases sont ensuite extraits à l'aide du modèle CRF présenté dans la partie précédente. La probabilité d'avoir une contradiction dans le document est également évaluée avec un classifieur de documents utilisant les représentations des deux phrases de contradiction choisies.

Lorsque les deux modules sont simplement empilés, cela crée un modèle avec un grand nombre de paramètres (environ 400M) et les deux tâches sont indépendantes alors que les paramètres et les contradictions sont intimement liés. Le choix des phrases devrait dépendre des résultats de l'exploration des paramètres. Par ailleurs, le résumé extractif atteint de bonnes performances mais il implique de modifier le fonctionnement habituel de l'encodeur en ajoutant des tokens de classification et une couche supplémentaire pour relier toutes les phrases. L'encodeur pré-entraîné doit donc être ré-entraîné après avoir ajouté ces paramètres. Il est ensuite affiné avec les données TRIZ. Le processus d'apprentissage est donc long et complexe. Le résumé abstraitif ne présente pas ces inconvénients, mais l'intégrité de l'information n'est plus garantie si les phrases sont modifiées. Pour contourner ce problème, nous proposons de détourner un modèle de résumé abstraitif vers le résumé extractif, c'est-à-dire que nous apprenons au réseau abstraitif à copier les phrases du document d'entrée. Le jeu de données pour le résumé extractif peut donc être réutilisé. De plus, ce modèle permet d'effectuer en même temps la fouille des paramètres, des contradictions et des principes inventifs sans ajouter de paramètres.

Le modèle multi-tâches montre des performances bien supérieures au modèle naïf avec notamment une amélioration de l'ordre de 15% en termes d'identifications de contradictions et paramètres.

Conclusion

Avec des réseaux de neurones de plus en plus puissants, notamment avec les architectures Transformers (DEVLIN et al., 2019; YANG et al., 2019; BELTAGY, PETERS et COHAN, 2020), l'automatisation du processus inventif qui est un enjeu majeur voit là une voie de progrès considérable. Dans cette thèse, notre objectif était de construire des modèles permettant une caractérisation des problèmes et solutions de brevets de tous domaines afin d'identifier des couples problèmes/solutions. Ces couples problèmes/solutions peuvent ensuite être exploités pour faciliter la résolution de nouveaux problèmes. Les problèmes sont identifiés sous la forme de contradictions permettant une formulation indépendante du domaine. Les solutions sont

identifiées via le principe inventif qui est à leur origine, c'est-à-dire la manière dont le problème a été résolu.

Le modèle multi-tâche développé ouvre de nouvelles possibilités notamment l'entraînement d'un modèle génératif à partir des contradictions et des principes inventifs pour générer des solutions.

Bibliographie

- BELTAGY, Iz, Matthew E. PETERS et Arman COHAN (2020). « Longformer : The Long-Document Transformer ». In : *arXiv :2004.05150*.
- CROCE, Danilo, Giuseppe CASTELLUCCI et Roberto BASILI (juill. 2020). « GAN-BERT : Generative Adversarial Learning for Robust Text Classification with a Bunch of Labeled Examples ». In : *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online : Association for Computational Linguistics, p. 2114-2119. DOI : 10 . 18653 / v1 / 2020 . acl - main . 191. URL : <https://www.aclweb.org/anthology/2020.acl-main.191>.
- DEVLIN, Jacob et al. (2019). « BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding ». In : *NAACL-HLT*.
- GOODFELLOW, Ian J. et al. (2014). « Generative Adversarial Nets ». In : *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'14. Montreal, Canada : MIT Press, 2672–2680.
- HERMANN, Karl Moritz et al. (2015). « Teaching Machines to Read and Comprehend ». In : *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'15. Montreal, Canada : MIT Press, 1693–1701.
- KIPF, Thomas N. et Max WELLING (2017). « Semi-Supervised Classification with Graph Convolutional Networks ». In : *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. ICLR '17. Palais des Congrès Neptune, Toulon, France. URL : <https://openreview.net/forum?id=SJU4ayYgl>.
- LIU, Yang et Mirella LAPATA (nov. 2019). « Text Summarization with Pretrained Encoders ». In : *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China : Association for Computational Linguistics, p. 3730-3740. DOI : 10 . 18653 / v1 / D19 - 1387. URL : <https://aclanthology.org/D19-1387>.
- SALIMANS, Tim et al. (2016). « Improved Techniques for Training GANs ». In : *Advances in Neural Information Processing Systems*. Sous la dir. de D. LEE et al. T. 29. Curran Associates, Inc. URL : <https://proceedings.neurips.cc/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf>.
- WESTON, Jason, Frédéric RATLE et Ronan COLLOBERT (2008). « Deep Learning via Semi-Supervised Embedding ». In : *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland : Association for Computing Machinery, 1168–1175. ISBN : 9781605582054. DOI : 10 . 1145 / 1390156 . 1390303. URL : <https://doi.org/10.1145/1390156.1390303>.
- YANG, Zhilin, William W. COHEN et Ruslan SALAKHUTDINOV (2016). « Revisiting Semi-Supervised Learning with Graph Embeddings ». In : *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA : JMLR.org, 40–48.
- YANG, Zhilin et al. (2019). « XLNet : Generalized Autoregressive Pretraining for Language Understanding ». In : *Advances in Neural Information Processing Systems 32*. Sous la dir. de H. WALLACH et al. Curran Associates, Inc., p. 5754-5764. URL :

<http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.pdf>.

Résumé

La théorie pour la résolution inventive des problèmes (TRIZ) a été proposée pour favoriser l'innovation de rupture et maximiser les chances de résoudre un problème de manière inventive. Altshuller, le fondateur de cette théorie, a présenté un pseudo-algorithme allant de la formulation d'un problème à sa résolution afin d'aider les ingénieurs. Sa théorie repose sur un constat très fort : chaque problème dans un domaine A peut être relié à un problème déjà résolu dans un domaine B, dont il suffirait d'adapter la solution. Cependant, trouver un problème correspondant dans un autre domaine nécessite, soit de lire toutes les publications scientifiques et brevets existants, soit de développer un algorithme capable de caractériser les problèmes résolus par chaque solution existante. Dans cette thèse, nous avons étudié cette deuxième option. Afin de permettre une recherche de solutions inter-domaines, des approches basées sur des réseaux de neurones sont développées pour identifier les concepts clés de la théorie TRIZ dans les brevets. En particulier, nous proposons une approche de fouille de contradictions basée sur le résumé automatique, une approche de fouille de paramètres intégrant la structure syntaxique des phrases et un modèle multi-tâches unifié.

Mots-clés : Traitement automatique du langage, Fouille de brevets, Réseau de neurones, Champ aléatoire conditionnel, TRIZ

Résumé en anglais

The Theory of Inventive Problem Solving (TRIZ) was proposed to push for disruptive innovation and maximize the chances of solving a problem in an inventive manner. Altshuller, the founder of this theory, presented a pseudo-algorithm going from the formulation of a problem to its resolution in order to help engineers. His theory is based on a very strong observation: each problem in a domain A can be linked to a problem already solved in a domain B, whose solution would only have to be adapted. However, finding a corresponding problem in another domain requires, either reading all the scientific publications and patents that exist, or developing an algorithm capable of characterizing the problems solved by each existing solution. In this thesis, we investigated this second research direction. In order to enable a cross-domain search for solutions, neural-based approaches are developed to identify the key concepts of TRIZ theory in patents. In particular, we propose a contradiction mining approach based on automatic summarization, a parameter mining approach integrating the syntactic structure of sentences and a unified multi-task model.

Keywords : Natural Language Processing (NLP), Patent Mining, Neural Network, Conditional Random Field, Theory of Inventive Problem Solving (TRIZ)