وزارة التعليم العالي والبحث العلمي

Abdelhafid Boussouf
University Center of Mila

Centre Universitaire

Abdelhafid Boussouf - Mila

المركز الجامعي

عبد الحفيظ بوالصوف – ميلة –

Institute of Science and Technology

Year 2023

Sector: Computer science

Thesis

Presented

In view of obtaining the Master's degree

# Un outil d'aide à l'annotation en vue de la construction d'un corpus annoté

Field: Mathematics and Computer Science

Sector: Computer science

Specialty: Information and Communication Sciences and Technologies (STIC)

By :

BOULDJADJ Badis

FEGHROUR Ishak

Board of the Jury

| | | |
|---|---|---|
| **President :** | BENHAMADA Sadek | U.C. Mila |
| **Supervisor :** | MEGUEHOUT Hamza | U.C. Mila |
| **Examiner:** | TALAI Meriem | U.C. Mila |

*Certainly, science guides, directs and saves;*

*Ignorance misleads, deceives and ruins*

Imâm Ali ibn Abi Talib

# الإهداء

بسم الله الرحمن الرحيم

أهدي تخرجي هذا إلى من علمني العطاء وإلى من أحمل اسمه بكل افتخار وأرجو من الله أن يمد في عمرك لترى ثمارا قد حان قطافها بعد طول انتظار "والدي العزيز"

وإلى ملاكي في الحياة وإلى معنى الحب والحنان والتفاني وإلى بسمة الحياة وسر الوجود وإلى من كان دعائها سر نجاحي أغلى الحبايب "امي الحبيبة"

وإلى من له الفضل الكبير في تشجيعي وتحفيزي ومن منه تعلمت المثابرة والاجتهاد وإلى من بهم أكبر وعليهم أعتمد وإلى من بوجودهم أكتسب قوة ومحبة لا حدود لها وإلى من عرفت معهم معنى الحياة "إخوتي وأخواتي"

وإلى من تحلوا بالإخاء وتميزوا بالوفاء والعطاء وإلى من برفقتهم في دروب الحياة السعيدة والحزينة سرت وإلى من كانوا معي على طريق النجاح والخير "أصدقائي الأعزاء"

وإلى أولئك المخلصين الذين لم يذخروا جهداً في مساعدتنا في مجال البحث العلمي، وأخص بالذكر الأستاذ المشرف على هذه الدراسة وصاحب الفضل في توجيهي ومساعدتي في إعداد هذه المذكرة، كما لا ننسى بالذكر أستاذة اللغة الانجليزية التي قامت بعملية التدقيق اللغوي، فجزاهما الله كل خير،

بتوفيق من الله، وبدُعاء من الام لم يبق سوى خطوات قليلة لإنهاء مسيرتي الدراسية، أهدي هذا العمل لكل من مد لي يد العون ولكل من نسيه قلمي ولم ينساه قلبي، واسأل الله التوفيق لي ولكم

**باديس**

# الإهداء

بسم الله الرحمن الرحيم

إلى من بلغ الرسالة وأدى الأمانة ونصح الأمة إلى نبي الرحمة ونور العالمين "محمد صلى الله عليه وسلم"

إلى التي بحنانها ارتويت وبدفئها احتميت وبنورها اهتديت أمي رحمها الله وجعلها تسكن بساتين النعيم

إلى درعي الذي به احتميت وفي الحياة به اقتديت إلى من علمني محاسن الاخلاق أبي اطال الله عمره

الى العائلة الكريمة الاخوة و الاخوات الى كل الاقارب

الى كل الأساتذة الرائعين الذين أثروا حياتي التعليمية في جميع اطواري الدراسية بمعرفتهم وخبراتهم

بالاخص الى الاستاذ المشرف حمزة مقحوت

إلى كل الأصدقاء والأحباب، إلى كل من يهمهم أمري إلى كل من أدركه القلب ولم يدركه القلم .

إليهم جميعا اهدي ثمرة هذا الجهد دعيا المولى إن ينفعنا بما علمنا ويعلمنا ما ينفعنا ويزيدنا علما.

**إسحاق**

# Acknowledgement

We thank Allah, who has helped us and granted us patience and courage during these long years of study. We would like to express our sincere gratitude to the professors who have assisted us in our journey, in addition to the success of this wonderful academic year. Our thanks primarily go to the faculty members and administrative staff in the Department of Mathematics and Computer Science for their rich and quality education, as well as their great efforts in providing modern training to their students. We would like to extend our heartfelt thanks to Mr. MEGUEHOUT Hamza, as our thesis supervisor, who has consistently shown attentiveness, inspiration, assistance, and time.

## Abstract

Artificial intelligence plays a crucial role in various fields, revolutionizing industries and enhancing human capabilities. One area where Artificial intelligence has demonstrated immense importance is in natural language processing. It enables computers to understand, interpret, and respond to human language, making it essential for applications such as virtual assistants, language translation, and sentiment analysis. Artificial intelligence powered natural language processing algorithms can analyze vast amounts of textual data, extracting valuable insights and enabling faster decision-making. With its ability to process and comprehend human language, artificial intelligence in natural language processing opens up new possibilities for improved communication, automation, and innovation across multiple domains.

Despite its significance, the Arabic language lacks comprehensive research. This hampers progress in natural language processing, machine translation, and language learning. Investing in Arabic language research would refine models, and preserve cultural richness.

For this reason, as Arabic speakers, we have chosen the field of Arabic natural language processing. Therefore, the objective of this thesis is to The objective is to utilize deep learning and machine learning techniques to create a semantic roles labeling tool for the Arabic language by training an artificial intelligence model.

For our project, we have selected the latest and largest database, OntoNotes 5.0, which is funded and supported by the US Department of Defense, the University of Pennsylvania, the University of Colorado and others.

**Key-words:** OntoNotes 5.0, Propbank, Semantic roles labeling, Natural language processing.

## Résumé

L'intelligence artificielle joue un rôle crucial dans divers domaines, révolutionnant les industries et renforçant les capacités humaines. Un domaine où l'intelligence artificielle a démontré une importance immense est le traitement du langage naturel. Il permet aux ordinateurs de comprendre, d'interpréter et de répondre au langage humain, ce qui est essentiel pour des applications telles que les assistants virtuels, la traduction de langues et l'analyse des sentiments. Les algorithmes de traitement du langage naturel alimentés par l'intelligence artificielle peuvent analyser de vastes quantités de données textuelles, extraire des informations précieuses et permettre une prise de décision plus rapide. Avec sa capacité à traiter et à comprendre le langage humain, l'intelligence artificielle dans le traitement du langage naturel ouvre de nouvelles possibilités pour une meilleure communication, une automatisation et une innovation améliorées dans de nombreux domaines.

Malgré son importance, la langue arabe souffre d'un manque de recherche approfondie. Cela entrave les progrès dans le traitement du langage naturel, la traduction automatique et l'apprentissage des langues. Investir dans la recherche sur la langue arabe permettrait de perfectionner les modèles et de préserver la richesse culturelle.

Pour cette raison, en tant que locuteurs arabes, nous avons choisi le domaine du traitement du langage naturel afin de le développer en arabe. Par conséquent, l'objectif de cette thèse est de former un modèle d'intelligence artificielle et de développer un outil pour l'étiquetage des rôles sémantiques de la langue arabe basé sur l'intelligence artificielle.

Pour notre projet, nous avons choisi la base de données la plus récente et la plus importante, OntoNotes 5.0, financée et soutenue par le département de la Défense des États-Unis, l'Université de Pennsylvanie, l'Université du Colorado et d'autres, en utilisant diverses méthodes d'intelligence artificielle telles que l'apprentissage en profondeur et l'apprentissage automatique.



**Mots-clés :** Étiquetage des rôles sémantiques, OntoNotes 5.0, Propbank, Traitement automatique du langage naturelle.

**الملخص**

يلعب الذكاء الاصطناعي دورًا مهمًا في مختلف المجالات ويحدث ثورة في الصناعات ويعزز القدرات البشرية. أحد المجالات التي أظهر فيها الذكاء الاصطناعي أهمية كبيرة هو معالجة اللغة الطبيعية. تمكن البرمجة اللغوية العصبية أجهزة الكمبيوتر من فهم اللغة البشرية وتفسيرها والاستجابة لها، مما يجعلها ضرورية للتطبيقات مثل المساعدين الافتراضيين وترجمة اللغة وتحليل المشاعر. يمكن لخوارزميات البرمجة اللغوية العصبية المدعومة بالذكاء الاصطناعي تحليل كميات هائلة من البيانات النصية، واستخراج رؤى قيمة وتمكين اتخاذ قرارات أسرع. بفضل قدرته على معالجة وفهم اللغة البشرية، يفتح الذكاء الاصطناعي في معالجة اللغة الطبيعية أفاقًا جديدة لتعزيز التواصل والتطوير التلقائي والابتكار عبر مختلف المجالات.

على الرغم من أهميتها، إلا أن اللغة العربية تفتقر إلى البحث الشامل. هذا يعيق التقدم في معالجة اللغة الطبيعية والترجمة الآلية وتعلم اللغة. الاستثمار في أبحاث اللغة العربية سيصقل النماذج ويعزز الشمولية ويحافظ على الثراء الثقافي.

لهذا السبب اخترنا كمتحدثي اللغة العربية مجال معالجة اللغة الطبيعية من أجل تطويرها في اللغة العربية. لذلك فإن الهدف من هذه الرسالة هو تدريب نموذج ذكاء اصطناعي وتطوير أداة لوصف الأدوار الدلالية للغة العربية على أساس الذكاء الاصطناعي.

بالنسبة لمشروعنا، اخترنا أحدث وأكبر قاعدة بيانات OntoNotes 5.0، التي تم إنشاؤها بتمويل ودعم من وزارة الدفاع الأمريكية، وجامعة بنسلفانيا، وجامعة كولورادو وغيرها، باستخدام أساليب مختلفة للذكاء الاصطناعي مثل التعلم العميق والتعلم الآلي.

**الكلمات المفاتحية:** تصنيف الأدوار الدلالية، OntoNotes 5.0، Propbank، معالجة اللغة الطبيعية.

# List of Abbreviation

| | |
|---|---|
| **API** | Application Programing Interface |
| **ATB** | Arabic Treebank |
| **AWN** | Arabic WordNet |
| **BSD** | Berkeley Software Distribution |
| **CBR** | Case-based reasoning |
| **DL** | Deep learning |
| **GUI** | Graphical User Interface |
| **IDE** | Integrated Development Environment |
| **K-NN** | K-nearest neighbor |
| **MSA** | Modern Standard Arabic |
| **NLP** | Natural Language Processing |
| **PWN** | Princeton WordNet |
| **QA** | Question answering |
| **SRL** | Semantic role labeling |
| **SVM** | Support vector machines |
| **WN** | WordNet |

# List of Figures

# List of Tables

# Table of Contents

# Introduction

# Introduction

Languages play a crucial role in human life, as they enable communication, preserve culture, foster connections and enhance cognitive abilities. In today's technologically advanced world, the rapid development of technology and science, coupled with the integration of computer science in our life, has necessitated the evolution of language-related fields. Recognizing its significance, the introduction of natural language processing has become essential.

Natural language processing is an area that gives great importance of understanding and analyzing human language. While this field has seen substantial progress in important languages like English, it still encounters challenges in languages such as Arabic.

Natural language processing is a collection of computational techniques for automatic analysis and representation of human languages [1], the Arabic language deserves an attention and a development in order to bridge the existing gap with other languages like English and foe unlock its potential.

Due to the limited scientific research in this language, There is just a few works [2] [3] [4], it has lagged behind in semantic role labeling compared to other languages. Therefore, it is crucial for the Arabic-speaking scientist community, particularly computer scientists, to invest more.

After conducting a bibliographic study, it has been observed that there is a deficiency in research concerning the annotation of semantic roles in Arabic, particularly in relation to the development of a larger textual resource. That is why in this thesis, our aim is to develop an annotation support tool for constructing an annotated Arabic corpus.

Utilizing deep learning and machine learning techniques, we have developed a support tool for semantic role labeling. This tool aims to aid annotators in assigning the appropriate class to constituents of the sentence by providing them with two predictions. These predictions are generated by deep learning and machine learning models.

We have divided our thesis into four chapters, with each one focusing on a specific aspect of our research.

The first chapter is titled "Semantic Role Labeling" introduce the field of study by a definition of semantic roles, the labeling process, formalisms and various annotation approaches. It finishes by discussing the systems of semantic role labeling in natural language processing.

We are interested in semantic role labeling for the Arabic language. Therefore, the second chapter is titled "Semantic Roles Labeling for the Arabic Language". This chapter begins with an overview of the

Arabic language, followed by an explanation of the meaning representation formalisms and it concludes by exploring works in semantic roles annotation in Arabic language.

In the third chapter (Deep Learning), we provide an overview of deep learning, covering fundamental concepts and notions. We then delve into application of neural networks in natural language processing. Finally, the chapter concludes with an examination of the use of deep learning in semantic role labeling.

The last chapter, titled "Implementation and Application," focuses on building our AI model, our experiences, discussions and developing the application. It begins by showcasing the tools utilized, followed by an exploration of the corpus used. We then proceed to build and train our deep learning model and implement it in our application.

# Chapter 1

# Annotation of Semantic Roles

# Chapter 1
# Semantic Roles Labeling

# 1- Introduction

We cannot discuss the development of an annotation support tool without addressing Semantic Role Labeling. That is why we begin with covering the fundamental concepts related to semantic roles.

This chapter starts with the definition of Semantic Roles, followed by the process of labeling these roles. It then introduces meaning representation formalisms, different annotation approaches, and concludes with an examination of the applications of semantic role labeling in natural language processing.

# 2- Semantic Roles

The representation of natural language sentences often incorporates the concept of semantic roles, which serve to link the grammatical components of a sentence to the underlying semantic representation. These roles are sometimes referred to as case roles, as they may be indicated by inflections in some languages (e.g., ablative in Latin). The term "thematic role" is used to refer to a role that captures the semantics of the sentence, such as event participation. The specific semantic roles used in Natural Language Processing (NLP) systems can vary widely, ranging from general roles like beneficiary to domain-specific roles like catalyst in chemistry [5].

Semantic roles, as discussed by [6], are labels used to describe recurring relationships between predicates and their associated arguments. These roles aim to capture differences and similarities in verb meanings that are reflected in argument expression. By identifying and generalizing these roles, we can contribute to the mapping from semantics to syntax. Thematic relations, theta roles, participant roles, and deep cases are alternative terms used to refer to semantic roles. Semantic roles are used to describe how the noun phrase refers to the participants, objects, or locations involved in the state, action, or situation described by the sentence. This provides a deeper understanding of the structure and meaning of language, allowing us to more accurately analyze and interpret the complex relationships between words and phrases in a sentence.

There are many semantic roles, including: [7]

**Agent**

Responsible for causing an action or change in another entity is known as the "agent role." ( the semantic role in a sentence that describes the argument which is responsible for causing an action that impact another entity).

Ex: *karim kicked the ball*

From the given sentence, "Karim" is likely performing the action that causes the ball to be kicked, making him the agent in this sentence.

**Patient**

A semantic role in a sentence that represents the entity that is affected by the action of another argument is referred to as the "patient role".

Ex: *The storm damaged the roof of the house.*

From the sentence, it can be inferred that the roof of the house was damaged as a result of the storm.

**Theme**

The theme typically refers to the participant or entity that undergoes a change of state or is affected by the action described by the verb. It is associated with verbs that involve transfer of possession, consumption, perception, or alteration of a state. The theme is generally considered to be a more general term that encompasses a broader range of semantic roles.

Example: "John ate an apple,".

The theme is "an apple" because it undergoes the action of being eaten".

In some linguistic frameworks, the terms "theme" and "patient" may be used interchangeably, considering them as synonymous roles. However, in certain frameworks or specific analyses, they may be distinguished based on the types of verbs or specific semantic properties associated with them.

**Location**

The semantic role in a sentence that identifies the location where the action took place is known as the "locative role".

Ex: *The battle was held in Ukraine.*

It is evident from the sentence that Ukraine was the location of the battle.

**Experiencer**

An argument that expresses the emotions or feelings of an individual can be referred to as an "experiencer role".

Ex: Anna sees the cat is happy.

This sentence shows that Anna saw the happiness of the cat.

**Instruments**

The semantic role that describes the means by which an action is performed in a sentence can be referred to as the "instrument role".

Ex: He played the guitar with his fingers.

This sentence shows that He played the guitar using his fingers.

**Goal**

The semantic role that describes the argument in a sentence indicating the place from which an action is directed is referred to as the "Goal role".

Ex: She sprayed the perfume from the bottle onto her wrist.

This sentence shows that the wrist is the place of the perfume must be.

**Source**

The role of an argument that describes the place from which an action originates is referred to as the "source role".

Ex: The sound of thunder came from the mountains.

This sentence shows that the source or the place of the sound of thunder comes from.

## 3- Semantic Roles Labeling

Semantic role labeling (SRL) is a technique used in natural language processing (NLP) to identify the relationship between the different parts of a sentence. İt specifically focuses on understanding the role that each word or phrase plays in relation to the main verb of the sentence. This role can include different parts like the actor, patient, or instrument of the verb, as well as other descriptors like location or time. By identifying that role, SRL can help improve the accuracy of various NLP tasks, such as language translation or sentiment analysis [8].

### 3.1- Terminology

There is too many concepts that refers to “Annotation of semantic roles” and for example:

- Semantic roles labeling [9].
- Frame Semantics [10].
- Deep Semantic Role Labeling [11].

Argument labeling [12].

## 3.2- The Semantic Roles Labeling (SRL) system

Semantic role labeling, the computational identification and labeling of arguments in text, has become a leading task in computational linguistics today.

The Semantic Roles Labeling goes through several steps [13]:

- Filtering (or pruning) the set of argument.
- Local scoring of argument.
- Apply a joint scoring (or global scoring).

Given a sentence and a designated verb, the SRL task consists of identifying the boundaries of the arguments of the verb predicate (argument identification) and labeling them with semantic roles (argument classification).The most common architecture for automatic SRL consists of the following steps to achieve these subtasks.

### 3.2.1- Filtering (or pruning) the set of argument

The initial phase of SRL involves reducing the number of potential argument candidates for a given predicate. Since these arguments can span multiple words or even disjoint sections of the sentence, any sequence of words in the sentence could be a candidate. SRL can improve the accuracy and efficiency of subsequent processing steps by filtering out unlikely options.

### 3.2.2- Local scoring of argument

The second step of the semantic role labeling process involves locally scoring the potential argument candidates using a function that assigns probabilities or confidence scores to each possible role label. An additional label is also included to identify candidates that should not be considered as arguments. During this step, each candidate is evaluated independently from the others. A key element in this scoring process is representing the candidates with features, rather than the specific choice of classification algorithm.

### 3.2.3- Apply a joint scoring (or global scoring)

The third and integral stage of Semantic Role Labeling involves utilizing joint scoring, also referred to as global scoring, to amalgamate the predictions of local scorers and establish a cohesive structure of labeled arguments for the predicate. This phase facilitates the identification of dependencies among various arguments of the same predicate.

Diverse adaptations of the three-step architecture have been identified, with some systems opting for local scoring exclusively or directly proceeding to joint scoring by skipping the intermediary stage. Additionally, a fourth step may be included to rectify common errors or ensure coherence in the final solution. This post-processing stage typically entails a series of handcrafted heuristic rules that are specific to the applied architecture and corpus [9].

# 4- Meaning Representation Formalisms

Lexical resources have been developed to enable the automatic processing of semantics in unfamiliar texts. These resources are the result of manual efforts, undertaken by teams with diverse objectives, applied to different types of data, and comprising individuals with varying intellectual backgrounds. Consequently, each resource offers distinct and valuable information about lexemes that is not available in others. The combination of these resources has the potential to enhance our knowledge of individual lexical items and their applications, particularly in concrete tasks such as question answering [14].

## 4.1- VerbNet

VerbNet [14], [15] a sophisticated system inspired by Levin's classes [16], offers a hierarchical arrangement of verb classes that organize verbs based on their syntactic and semantic properties.

Each class is defined by sets of verbs and lists of arguments, along with syntactic and semantic information.

Thematic roles and binary predicates describe the arguments, while syntactic information maps them to deep-syntactic arguments, accounting for voice alternations and transformations.

Additionally, semantic predicates capture the nuances of participants during different stages of the event described by the verb's syntactic frame.

VerbNet has been extended from the original Levin classes, now encompassing an impressive array of senses and lexemes.

Its primary objective is to facilitate the acquisition of new class members by grouping verbs into classes with coherent syntactic and semantic characteristics. The hierarchical structure and limited number of thematic roles in VerbNet support generalizations, making it a valuable tool for computational linguistics and natural language processing research.

In VerbNet [17] each verb class is defined in its entirety by a specific group of members that share common characteristics and features .the thematic roles dictating the predicate-argument structure of these members, the selectional constraints imposed on the arguments, and the frames encompassing both a syntactic depiction and semantic predicates that serve a temporal function.

A summary of how this integration affected VN and the result of the extended VN is shown in Table [1.1].

Table 1.1: Summary of the Lexicon's Extension [18]

| | Original VN | Extended VN |
|---|---|---|
| **First-level classes** | 191 | **274** |
| **Thematic roles** | 21 | **23** |
| **Semantic predicates** | 64 | **94** |
| **Syntactic restrictions (on sentential compl)** | 3 | **55** |
| **Number of verb senses** | 4656 | **5257** |
| **Number of lemmas** | 3445 | **3769** |

## 4.2- PropBank

PropBank is a project aimed at providing training data for automatic role labelers. To accomplish this, PropBank uses the same corpus as the Penn Treebank project [19], which provides manually verified syntactic parse trees for every data point. This feature makes PropBank a valuable practical resource for natural language processing tasks. In an effort to understand the significance of syntax in semantic-role labeling, researchers compared the accuracy of a system based on error-free parses with one using parser output generated automatically. Additionally, they tested whether mistakes made by automatically generated parsers negate the supplementary details found in complete parse trees by examining a labeling system that uses a condensed or 'chunked' representation of the input. Through these analyses, the researchers were able to gain insights into the suitability of PropBank for its stated purpose and the role of syntax in natural language processing [20].

In various instances of a verb, labels assigned to arguments must be consistent, even when the verb appears in different syntactic roles. This means that the same label should be assigned to the same argument, even if it appears in different syntactic roles. For example, if the argument labeled as Arg1 in the sentence "John broke the window" refers to the same window that is labeled as Arg1 in "The window broke", then the label assigned to the argument should be consistent, regardless of the syntactic role it occupies. Consistency in argument labeling is necessary for accurate semantic role labeling, as it ensures that the meaning of the argument is preserved despite variations in its syntactic role.

PropBank's primary objective is semantic role annotation using consistent and universal labeling. This project aims to annotate a significant amount of meaningful text that is suitable for supervised machine learning models to learn from. By providing accurate and reliable annotation of semantic roles, PropBank can make a valuable contribution to the field of natural language processing and improve the accuracy of various natural language understanding tasks.

PropBank is a valuable resource that provides data on the frequency of word usage for statistical analysis and generation components for natural language applications. In addition, PropBank contains a lexicon that classifies each word into broad senses called "framesets." Each frameset includes information about the argument roles that can be associated with the corresponding word and provides examples of usage in various syntactic contexts. The lexicon is used as a reference for annotators to define semantic roles for verbs. Although the lexicon is similar to FrameNet, it is more general-purpose in its specifics and more coarse-grained [14].

### 4.3- FrameNet

The FrameNet project, which has been operational since 1997 and is supported by the National Science Foundation, is building a lexical database of English. This database is designed to be both human- and machine-readable, and contains over 13,000 word senses with annotated examples that showcase their meaning and usage. As a result, it serves as a valuable resource for students and teachers of linguistics, providing detailed evidence for the combinatorial properties of English vocabulary.

Moreover, the manually annotated sentences in FrameNet, which are linked to more than 1,200 semantic frames, serve as a unique training dataset for semantic role labeling. This has a wide-ranging applications in Natural Language Processing, including information extraction, machine translation, event recognition, sentiment analysis, and more. As such, FrameNet is a valuable tool for researchers in the field of Natural Language Processing, offering a rich source of data for training and development.

Furthermore, the availability of FrameNet data for free download has made it widely accessible to researchers around the world, who have utilized it for diverse purposes. Additionally, FrameNet-like databases have been developed for other languages, and efforts are underway to align FrameNets across languages. This underscores the global impact and potential of FrameNet in advancing linguistic research and supporting language-related applications in various contexts [21].

## 4.4- Semantic roles in different projects

Table [1.2] lists Semantic roles in different projects.

Table 1.2: Semantic roles in VerbNet, PropBank and FrameNet [22]

| VerbNet | PropBank | FrameNet |
|---|---|---|
| Agent | Arg0, Arg1 | Agent, Speaker, Cognizer, Communicator, Ingestor, Deformer, etc. |
| Actor | Arg0 | Avenger, Communicator, Item, Participants, Partners, Wrongdoer |
| Actor1 | Arg0 | Arguer1, Avenger, Communicator, Interlocutor1, Participant 1, etc. |
| Actor2 | Arg1, Arg2 | Addressee, Arguer2, Injured Party, Participant2, Partner2 |
| Attribute | Arg1, Arg2 | Attribute, Dimension, Extent, Feature, etc. |
| Beneficiary | Arg1, Arg2, Arg3, Arg4 | Audience, Beneficiary, Benefitted party, Goal, Purpose, Reason, Studio |
| Cause | Arg0, Arg1, Arg2, Arg3 | Addressee, Agent, Cause, Communicator, etc. |
| Destination | Arg1, Arg2, Arg5 | Addressee, Body part, Context, Goal, etc. |
| Experiencer | Arg0, Arg1 | Cognizer, Experiencer, Perceiver, etc. |
| Extent | Arg2 | Difference, Size change |
| Instrument | Arg2 | Agent, Fastener, Heating instrument, Hot Cold source, etc. |
| Location | Arg1, Arg2, Arg3, Arg4, Arg5 | Action, Area, Fixed location, etc |
| Material | Arg1, Arg2, Arg3 | Components, Ingredients, Initial entity, Original, Resource, Undergoer |
| Patient | Arg0, Arg1, Arg2 | Addressee, Affliction, Dryee, Employee, Entity, Executed, etc. |
| Patient1 | Arg0, Arg1 | Concept 1, Connector, Fastener, Item, Item 1, Part 1, Whole patient |
| Patient2 | Arg2, Arg3 | Concept 2, Containing object, Item 2, Part 2 |
| Predicate | Arg1, Arg2 | Action, Category, Containing event, etc. |
| Product | Arg1, Arg2, Arg4 | Category, Copy, Created entity, etc. |
| Proposition | Arg1, Arg2, | Act, Action, Assailant, Attribute, etc. |
| Recipient | Arg1, Arg2, Arg3 | Addressee, Audience, Authorities, Recipient |
| Stimulus | Arg1 | Emotion, Emotional state, Phenomenon, Text |
| Theme | Arg0, Arg1, Arg2 | Accused, Action, Co-participant, Co-resident, Content, Cotheme, etc. |
| Theme1 | Arg0, Arg1 | Cause, Container, Phenomenon 1, Profiled item, Theme |
| Theme2 | Arg1, Arg2, Arg3 | Containing object, Contents, Cotheme, etc. |

| | | |
|---|---|---|
| Time | ArgM TMP | Time |
| Topic | Arg1, Arg2 | Act, Behavior, Communication, Content, etc. |
| Asset | Arg1, Arg3 | Asset, Category, Measurement, Result, Value |
| Value | Arg1 | Measurement, Result, Value, Asset, Category |
| Source | Arg2, Arg3 | Role, Victim, Patient, Source, Path start, etc. |
| - | - | Setting, ContainingEvent |
| - | - | Means |
| - | ArgM _Manner | Manner |
| - | ArgM _Purpose | Purpose |

## 5- Annotation Approches

In [23] the semantic role labeling systems use two types of resources:1- Inventories2- Annotated corpora. These resources are used differently depending on methods, which are often divided into four general approaches: supervised, knowledge-based, semi-supervised, and unsupervised [24].

### 5.1- Supervised

The supervised methods [24], [25], [26] and [27] use an annotated corpus and therefore adopt the associated inventory. Classical machine learning techniques are used to determine the correct meaning of each occurrence of a word given the information obtained from the context of that occurrence. Supervised semantic role annotation is often divided into several sub-tasks, sometimes partially grouped.

- Identification of predicates.
- Identification of frames.
- Identification of arguments that establish the phrases playing a role in the sentence.
- Classification of roles that determines the actual role of each phrase among those selected in the previous phase.

### 5.2- Based on knowledge

Some approaches do not use annotated corpora but rely on the VerbNet resource [28][29].These systems then free themselves from the small size inherent in any annotated corpus and rely on subcategorization frames for annotation. A sense inventory is used: classification still needs to be done, and the main difficulty here is to obtain useful information without annotated examples. Since these methods continue to use an inventory, it is still possible to compare results between different systems and perform an evaluation on a ground truth.

### 5.3- Semi-supervised

The author in [30]annotated the English Europarl corpus with an automatic system trained on PropBank, and then used the corpus alignments to obtain a French corpus automatically annotated in PropBank roles. This is an interesting approach to obtain semantically annotated corpora for French. Various difficulties are remained: the English PropBank data was directly used for French, and the scores are still too low, requiring manual correction of the corpus. Similarly, In [31]the author propose to automatically develop a new PropBank for Swedish based on English PropBank and using Wikipedia as a parallel corpus. The accuracy of the obtained data is still unknown.

### 5.4- Unsupervised

These approaches do not use any prior knowledge, whether it be an inventory or an annotated corpus. An unsupervised approach must necessarily construct its own inventory. This can be done through sense clustering based on context occurrences found in the corpus [32][33][34][35].The potential benefits are numerous. These algorithms require no resources and offer two interesting properties:

- The chosen inventory closely matches the corpus used, which allows it to avoid overly fine distinctions and adapt to new domains via new corpora, as the domain has a significant impact on the senses used.
- The more text is available, the more effective system can become.

## 6- The Use of Semantic Roles in Natural Language Processing

There are many projects that have used semantic roles in the field of natural language processing. In this section, we will review some of those works.

### 6.1- Semantic roles and machine translation

According to [36],the authors in [37]observed that semantic roles exhibit greater consistency across languages compared to syntactic constituents. other authors [38] viewed this feature as a driving force behind the approach of using consistency of semantic roles as a criterion for selecting machine translation outputs.

There is a lot of work in the field of machine translation. We can suggest some [21] [22].

### 6.2- Semantic roles and automatic summarization

Based on the research of [41] the value of semantic roles in summarization was examined for the first time by Kristofferson as documented [42]. In a more recent study, Suanmali and other [43]made use of semantic roles and WordNet (Fellbaum, 1998) to calculate the semantic similarity between two sentences, enabling the determination of whether or not sentences should be included in the abstract.

in this paper [41]they proposed method combining semantic roles and named entity for sentence extraction.

The architecture of the proposed method is presented in Figure [1.1].

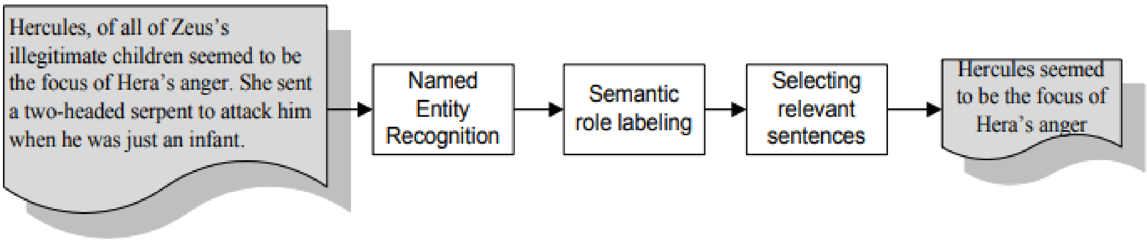



Figure1.1:Overall architecture of the proposed summary generation method based on semantic roles [41]

### 6.3- Semantic roles and question/answer systems

Open domain question answering (QA) involves leveraging the vast expanse of the World Wide Web as a source of data to extract succinct responses to natural language queries [44].

Historically, question answering systems have relied on a diverse set of lexical resources to overcome surface discrepancies between questions and their corresponding answers.

In the realm of answering complex questions, Narayanan and Harabagiu were pioneers in emphasizing the critical role of semantic roles. Their approach involves merging semantic role information derived from PropBank and FrameNet to identify predicate argument structures. The system then utilizes probabilistic inference over these structures, combined with a topic model specific to the domain, to extract anticipated answers [45].

Modern QA systems leverage large text collections to extract answers by:

- Predicting the type of answer they expect.
- Ranking potential answers to pinpoint the exact one.
- Employing question-related keywords or patterns to identify answer passages.

By employing semantic extraction, it becomes possible to recognize predications in the input text. To tackle intricate questions, the question class or pattern is determined in addition to relevant sections of the scenario. These relevant segments are referred to as the topic model [46].

In this paper [45],the authors show us the impact of semantic roles in QA and whether they help.

## 7- Conclusion

This chapter provided a comprehensive introduction to semantic role labeling by exploring the main concepts of semantic roles. It covered various aspects, including the definition of semantic roles and the practical applications of semantic role labeling in natural language processing.

In the next chapter, we will focus specifically on semantic role labeling in the Arabic language.

# Chapter 2 Semantic Roles Labeling for Arabic language

# Chapter 2

# Semantic Roles Labeling for Arabic Language

# 1 - Introduction

Since our study focuses on the annotation of semantic roles in the Arabic language, the second chapter delves into semantic roles labeling in the Arabic language. It start with an overview of the Arabic language, followed by an explanation of the meaning representation formalism tailored for Arabic. The chapter concludes by exploring various significant works on semantic roles in the Arabic language and their results.

# 2 - Arabic language

Arabic is a widely spoken and written language with over 300 million speakers worldwide, and it holds the status of an official language within the United Nations. It is also the language of the Quran, which is read by a significant portion of the global Muslim community. Furthermore, Modern Standard Arabic (MSA) serves as the shared formal language of educated individuals within the Arab world, our focus is solely on MSA.

When considering its morphology, Arabic can be described as having a rich system. Similar to English, Arabic verbs contain explicit markings for tense, voice, and person. However, Arabic verbs also possess additional markings for mood, including subjunctive, indicative, and jussive. In the realm of nominal words (such as nouns, adjectives, proper names, and pronouns), Arabic marking includes features such as case (accusative, genitive, nominative), number, gender, and definiteness [47].

# 3- Meaning representation formalisms for Arabic

## 3.1- VerbNet

The Arabic VerbNet is an extensive verb lexicon that employs Levin's classes and the developmental procedure of Kipper Schuler to provide comprehensive syntactic and semantic information about Arabic verbs. The current VerbNet contains 202 classes that encompass 4707 verbs and 834 frames. Each class is structured hierarchically and comprises tuples that contain essential information about the verb, including its root form, deverbal form, and participle. Moreover, thematic roles and their constraints are also encoded at the same level. The frames within each class reflect alternations where the verbs can appear and are represented as example sentences, syntactic structures, and semantic structures that incorporate semantic predicates and their arguments. Additionally, temporal information is presented in a manner similar to Moens and Steedman (1988), which makes the Arabic VerbNet a valuable resource for researchers in Arabic language processing [48].

### 3.2-WordNet

Princeton WordNet (PWN) for English got good results and was described as successful in this article [49],As a result it motivated researchers to develop and create WordNet for other languages .

The creation of the Arabic WordNet (AWN) project was based on the methods of Princeton WordNet and EuroWordNet.

The result of this endeavor was a linguistic and semantic resource, according to WordNet (WN) architecture, respecting some unique aspects of the Arabic language, including pronunciation, (irregular) plural (BP), and roots, AWN also includes some cultural terms [50].

Table [2.1] shows Comparison of AWN content with an Arabic lexicon and other WNs .

Table 2.1: Comparison of AWN content with an Arabic lexicon and other WNs [50]

| Figures | Arabic | Spanish | English |
|---|---|---|---|
| **W N synsets** | 9,698 | 57,424 | 117,659 |
| **W N word-senses** | 18,925 | 106,566 | 206,941 |
| **W N word lemmas (WL)** | 11,634 | 67,273 | 155,287 |
| **Language lemmas (LL)** | 119,693 | 104,000 | 230,000 |
| **Ratio lemmas (WL/LL) (%)** | 9.7 | 64.7 | 67.5 |
| **Ratio word-lemmas (WN/English WN) (%)** | 7.5 | 43.3 | 100.0 |
| **Ratio synsets (WN/English WN) (%)** | 8.2 | 48.8 | 100.0 |
| **Ratio word-senses (WN/English WN) (%)** | 9.1 | 51.5 | 100.0 |

### 3.3- FrameNet

In the official FrameNet site [51], we see that there are resources for French, Chinese, German, etc., but no resources are available for Arabic.

Actually, there are serious attempts underway to create an Arabic FrameNet. Among these attempts, we would like to mention the work of [52]. This work is considered an introductory step for Arabic FrameNet, but it is currently still in an early version.

## 3.4- TreeBank

The Penn Arabic Treebank (ATB) project was initiated in the autumn of 2001,has completed three releases of morphologically and syntactically annotated data. These include [53]:

- Arabic Treebank: Part 1 v 2.0, LDC Catalog No. LDC2003T06, consisting of approximately 166K words of written Modern Standard Arabic newswire from the Agence France Presse corpus.
- Arabic Treebank: Part 2 v 2.0, LDC Catalog No. LDC2004T02, containing roughly 144K words from Al-Hayat distributed by Ummah Arabic News Text. The UMAAH corpus features new annotations such as complete vocalization, including case endings, lemma IDs, and more specific part-of-speech tags for verbs and particles.
- Arabic Treebank: Part 3 v 1.0, LDC Catalog No. LDC2004T11, which consists of around 350K words of newswire text from An-Nahar, with 150K of those words having been treebanked in ATB: Part 3(a) v 1.1 (LDC2004E71) and morphologically annotated.

## 3.5- Propbank

The process for constructing the Arabic propbank shares a strong resemblance to the design methodologies employed for prior languages. The overarching approach centers on developing framesets for verbs, which serve as a blueprint for annotators to follow when performing annotations. These framesets are instrumental in identifying the predicate and its potential arguments.

The PropBank annotation framework relies on a foundational syntactic structure to establish semantic annotations.

In Arabic propbank, the authors in utilized the Arabic Treebank (ATB) constituency parses for the underlying syntax [47].

Hence in this example [47]:

مشروع الامم المتحدة فرض مهلة نهائية ل اتاحة الفرصة امام قبرص

m$rwE AlAmm AlmtHdp frD mhlp nhAyp l AtAHp AlfrSp AmAm qbrS.

***Meaning:*** The United Nations' project imposed a final grace period as an opportunity for Cyprus

***Is annotated as follows*:**

[m$rwE AlAmm AlmtHdp] → ARG0

[frD] → PREDICATE

[mhlp nhA}yp] → ARG1

[l AtAHp AlfrSp AmAm qbrS] → ARGM−PRP

ARG0 corresponds to the subject of the sentence which is m$rwE AlAmm AlmtHdp 'United Nations project'; ARG1 corresponds to the object position, namely, mhlp nhAyp 'final grace period'. The predicate has an ARGM-PRP (purpose argument) in l AtAHp AlfrSp AmAm qbrS 'as an opportunity for Cyprus'.

Table 2.2: Meaning and arguments PropBank

| Tag | Describtion |
|---|---|
| ARG0 | Agent, operateur |
| ARG1 | Thing operated |
| ARG2 | Explicit patient (thing operated on) |
| ARG3 | Explicit argument |
| ARG4 | Explicit instrument |

## 4- SLR system for Arabic

Mona Diab and others in [2] proposed the first semantic role labeling (SRL) system for a Semitic language, specifically the Arabic language. The underlying concept behind the system was to investigate the potential of adapting the technology developed for English to Arabic. The methodology involved utilizing a supervised model that leverages support vector machines (SVM) technology for both argument boundary detection and argument classification.They have used the dataset SEMEVAL 2007 Task 18 Characterized by:

- Based on Arabic PropBank and TreeBank.
- Covers 95 verbs from TreeBank.
- 886 sentences for development and 1,725 arguments.
- 902 sentences for the test and 1,661 arguments.
- 8,402 sentences for training and 21,194 arguments.

In their experimental approach, sought to determine the efficacy of applying the technology proposed in prior research for automatic SRL of English texts to Arabic SRL systems.

The evaluation process employed by Mona Diab et al. entailed separate testing of each phase of the SRL procedure, namely boundary detection and argument classification.

In this first contribution, the results are 94.06% for detection and 81.43% for classification.

The same researchers present another [54] work which takes advantage of the rich morphological features of the language, which uses support vector machines and kernel methods. Their work is piloted on Arabic Propbank data. It achieved good results at 82.17% and it was a huge improvement.

In [3], the authors presented a new method for semantic role labeling in Arabic sentences, leveraging the case-based reasoning approach (CBR) in conjunction with the powerful supervised K-nearest neighbor (K-NN) method to achieve superior results. They used a newer and larger dataset, OntoNotes 5.0, compared to the one used in the previous work, and achieved an accuracy of 62.42% on argument classification.

The same researchers hybridized their previous method with deep learning (DL) in [4] (using CBR K-NN, and DL) and achieved an even better result of 88.66%.

In 2022, another study [55] on the semantic roles Labeling in Arabic was conducted, using the same OntoNotes 5.0 database and artificial intelligence methods , including K-nearest neighbors and deep learning. The researchers conducted several experiments and obtained the following results: the first experiment achieved a 73.66% accuracy using machine learning, while the second experiment, which used random forest, achieved an accuracy of 77.42%. The best result was obtained using deep learning, with an accuracy of 81.31%.

## 5- Conclusion

In this chapter, we have presented an overview of the characteristics of the Arabic language and discussed the key forms of meaning representation used in Arabic. We decided to introduce deep learning in the next chapter.

# Chapter 3

# Deep learning

# Chapter 3

# Deep Learning

# 1- Introduction

Based on the results of other Arabic SRL systems mentioned in the previous chapter, it is evident that deep learning is the most effective technology for semantic role labeling (SRL).

Therefore, we have dedicated the third chapter to deep learning. Firstly, we discuss neural networks, followed by an exploration of deep learning and architectures of artificial neural networks. We then delve into the training of neural networks and their application in natural language processing. The chapter concludes with an examination of deep learning in semantic role labeling.

# 2- Deep learning

Deep Learning techniques have recently emerged as a field of research aiming to bridge the gap between Machine Learning and Artificial Intelligence. By enabling the learning of multiple levels of representation and abstraction, Deep Learning provides powerful tools to extract meaningful insights from complex data, including images, audio, and natural language [56].

Deep Learning is a category of machine learning methods that leverage multiple layers of hierarchical supervised architectures for unsupervised feature learning, as well as pattern analysis and classification. At the core of deep learning lies the ability to calculate hierarchical features or representations of data, where higher-level factors or features are constructed from lower-level ones. The suite of deep learning techniques has been rapidly expanding and now encompasses a neural network architectures, hierarchical probabilistic models, as well as supervised and unsupervised feature learning algorithms [57].

# 3- Neural network

Neural networks have emerged as a potent set of tools for addressing challenges in pattern recognition, data analysis, and control. One of the most significant advantages of neural networks is their impressive processing speeds [58].

Artificial neural networks possess a significant attribute wherein they can acquire knowledge from presented samples or patterns, thereby capturing the system's behavior. Consequently, once the network comprehends the connection between inputs and outputs, it gains the ability to generalize solutions. This implies that the network can generate outputs that closely approximate the expected or desired outputs for a wide range of input values [59].

From the point of view of researchers in [58] neural network can be viewed as a non-linear mathematical function that maps input variables to output variables using a set of parameters known as weights. These weights can be determined through a process of learning or training, which typically involves using a set of examples. Although this process can be computationally intensive, once the weights have been determined, the network can rapidly process new data.

## 4- Architectures of Artificial Neural Networks

In the context of artificial neural networks, the system can typically be divided into three key components As shown in the figure [3.1], each of which is referred to as a layer. These layers are [59]:

- The input layer plays a crucial role in obtaining diverse forms of external input, whether it be data, signals, features, or measurements, as a preliminary step in the complex process of information processing.
- The hidden layers consist of a collection of neurons that are tasked with extracting intricate patterns relevant to the process or system under scrutiny. These layers undertake the biggest share of the network's internal processing, playing a crucial role in the overall performance of the system.
- The output layer is a collection of neurons that assume the critical responsibility of generating and delivering the ultimate network outputs. These outputs result from the intricate processing executed by the neurons in the preceding layers.

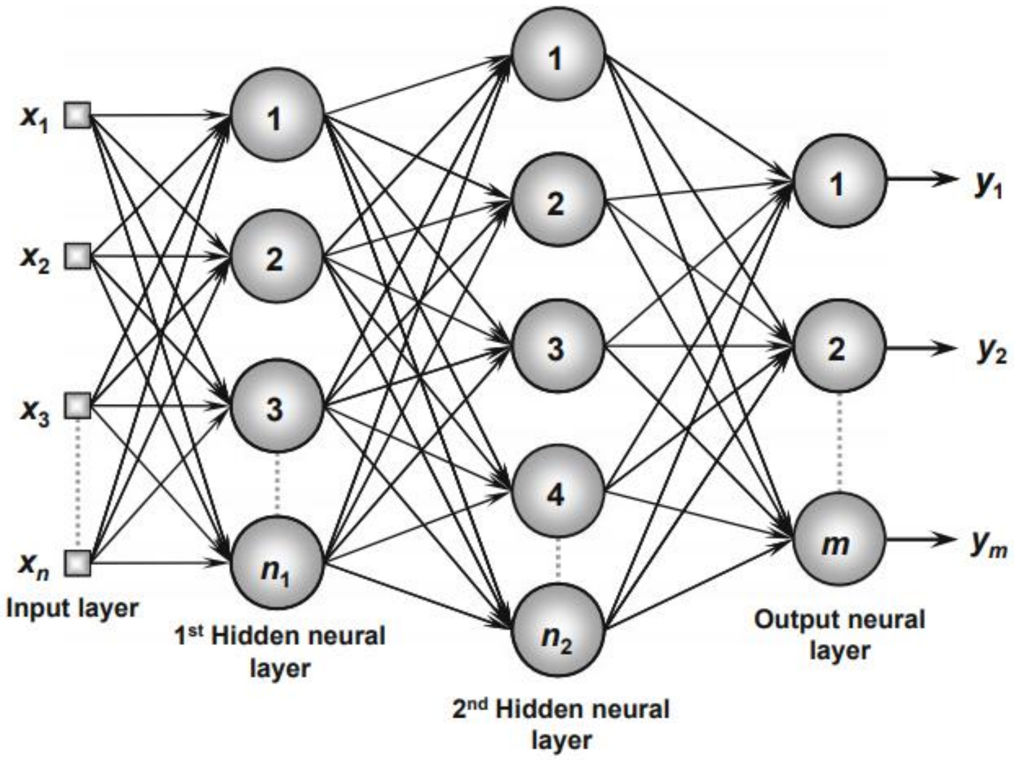



Figure 3.1 Example of a feedforward network with multiple layers [59]

# 5- Training neural networks

The process of training a neural network involves carefully adjusting the synaptic weights and thresholds of its neurons, in a systematic and organized manner, to enable the network to produce accurate solutions that can be generalized. This step-by-step process is known as the learning algorithm. Through this process, the network is able to extract relevant and discriminating features from samples that are collected from the system being mapped.

To ensure that the network is robust and can produce accurate results, the complete set of available samples is typically divided into two subsets: the training subset and the test subset. The training subset consists of randomly selected samples, comprising between 60-90% of the complete set, and is primarily used in the learning process. On the other hand, the test subset is composed of between 10-40% of the complete sample set, and is used to verify whether the network's ability to generalize solutions is satisfactory, thus facilitating the validation of a given topology [59].

## 5.1 Activation functions in keras

Keras provides a wide range of activation functions for building neural networks. Activation functions are used to introduce non-linearity into the neural network, allowing it to learn complex patterns and make predictions. Without activation layers, neural networks would be linear functions, and would not be able to learn complex relationships between inputs and outputs. These are some of the activation functions we used in our work [60]:

- The **ReLU** activation function is a piecewise linear function that returns 0 for negative inputs and x for positive inputs. The ReLU function is very efficient to computer, and it has been shown to be very effective for training deep neural networks.
- The **softmax** activation function is a non-linear function that normalizes a vector of values to a probability distribution. The softmax function is often used for the output layer of a neural network, where the output represents the probability of each class.
- The **LeakyReLU** activation function is a variation of the ReLU function that has a small positive slope for negative inputs. This helps to prevent the problem of "dead neurons" that can occur with ReLU, where neurons that are initialized with negative weights never activate and do not contribute to the output of the neural network.

- The **Tanh** activation function is a hyperbolic tangent function that has a range of [-1, 1]. This what makes it useful for tasks where the output of the neural network should be a real number between -1 and 1, such as regression tasks.

# 6- Machine learning

## 6.1- Definition

Machine learning is an evolving branch of computational algorithms that are designed to emulate human intelligence by learning from the surrounding environment. They are considered the working horse in the new era of the so-called big data. Techniques based on machine learning have been applied successfully in diverse fields ranging from pattern recognition, computer vision, spacecraft engineering, finance, entertainment, and computational biology to biomedical and medical applications [61].

## 6.2- Machine learning VS Deep learning

The table [3.1] shows the main different between machine learning and deep learning

Table 3.1: The main different between machine learning and deep learning [62]

| Machine learning | Deep learning |
|---|---|
| A subset of AI | A subset of machine learning |
| Can train on smaller data sets | Requires large amounts of data |
| Requires more human intervention to correct and learn | Learns on its own from environment and past mistakes |
| Shorter training and lower accuracy | Longer training and higher accuracy |
| Makes simple, linear correlations | Makes non-linear, complex correlations |
| Can train on a CPU (central processing unit) | Needs a specialized GPU (graphics processing unit) to train |

# 7 - Deep learning in natural language processing

Natural language processing has made significant strides thanks to its use of corpus, lexicon databases, and neural networks. In particular, the application of deep learning methods has enabled artificial neural networks to non-linear process, leading to more accurate and valuable NLP tools These advancements have brought about a paradigm shift in the field. Multi-layer neural networks have further expanded the capabilities of NLP, allowing for faster and more reliable processing of natural language [63].

In the paper referenced as [62], the authors discuss the application of (DL) in machine translation. They introduce two distinct categories of DL approaches for machine translation.

On the other hand, the paper referenced as [63] focuses on the utilization of DL techniques in two areas: question answering (QA) over knowledge bases and machine comprehension. The authors delve into the application of DL methods specifically in these domains.

## 8 - Deep learning in semantic roles labeling

Semantic role labeling, which involves the computational identification and labeling of arguments in text, has emerged as a prominent task within computational linguistics today. While researchers have explored this topic for several decades, recent advancements in statistical machine learning techniques, coupled with the availability of substantial resources, have spurred a surge of interest and effort in this field [64].

There are many projects that worked on semantic roles using deep learning, In [65], the authors introduced a new deep learning model that utilized an 8-layer BiLSTM highway architecture, recurrent neural networks with dropout, and highway connections. Additionally, they employed the A* decoding algorithm. These factors were crucial in achieving promising results. They applied the model to CoNLL 2005 and CoNLL 2012 datasets and observed a significant reduction in the relative error of 10% compared to previous state.

## 9 – Conclusion

In this chapter, we have presented an overview of the fundamental principles of deep learning and highlighted some of the work conducted in semantic role labeling using deep learning. In the final chapter, we will present our application and the results of experiments conducted using deep learning techniques and machine learning.

# Chapter 4 Implementation and Application

# Chapter 4

# Implementation and Application

# 1 - Introduction

In this chapter, we will discuss our implementation and the developed tool. The first part begins by presenting the tools used during the development process and provides an exploration of the used corpus. In the second part, we will present our proposed models, describe the experiments, and discuss the obtained results. Finally, we will utilize these models to construct our tool.

# 2 - Development tools

We have used many tools in our research, including the Python as programming language, Pycharm[1] as an Integrated Development Environment (IDE), and ChatGPT[2] to ask about some libraries or instructions, especially during the development of the Graphical User Interface (GUI).

## 2.1- Python programing language

Python is a programming language that offers dynamic semantics and operates through interpretation. It follows an object-oriented approach and is considered a high-level language. Python's appeal lies in its built-in data structures at a high level, which, when combined with dynamic typing and binding, make it highly suitable for Rapid Application Development. Additionally, it serves well as a scripting or glue language to connect various existing components. Python's syntax is simple and easy to grasp, prioritizing readability and minimizing the expenses associated with program maintenance. The language supports modules and packages, fostering program modularity and facilitating code reuse. Python's interpreter, along with its extensive standard library, is freely available in both source and binary formats for all major platforms, and can be distributed without any charge [66].

The Python programming language is widely regarded as the most effective tool for automating tasks due to its superior simplicity and consistency compared to other programming languages. Additionally, the vibrant Python community provides a supportive environment for developers to exchange ideas and collaborate on improving their code [67].

Python offers an extensive range of libraries and frameworks, greatly simplifying the coding process and saving valuable time. Among the widely used libraries are NumPy, designed for scientific calculations, SciPy for more intricate computations, and scikit for data mining and analysis. These libraries seamlessly integrate with robust frameworks such as TensorFlow, CNTK, and Apache Spark, playing a crucial role in machine learning and deep learning endeavors [68].

[1] www.jetbrains.com/pycharm/
[2] chat.openai.com

In the TIOBE index [69] (updated May 2023), Python is the most popular programming language, followed by C and java. Python's population has increased by 0.71% this year, and it cover 13.45% of the population. This is shown in the figure [4.1]:

| May 2023 | May 2022 | Change | Programming Language | Ratings | Change |
|---|---|---|---|---|---|
| 1 | 1 | | Python | 13.45% | +0.71% |
| 2 | 2 | | C | 13.35% | +1.76% |
| 3 | 3 | | Java | 12.22% | +1.22% |
| 4 | 4 | | C++ | 11.96% | +3.13% |
| 5 | 5 | | C# | 7.43% | +1.04% |

Figure 4.1: The most popular programming language – TIOBE index [69]

In our work we used Keras[1], Numpy[2], Pandas[3], Scikit-learn[4], Tensflow[5] and Tkinter[6] to train the model and test it in the GUI application. Next, we clarify the use of these libraries in our work

## 2.2- Pycharm

PyCharm is an IDE designed specifically for Python developers. It offers a comprehensive set of essential tools that are seamlessly integrated, resulting in a convenient and productive environment for Python, web, and data science development.

- **Community** (available as a free and open-source solution) provides a comprehensive suite of tools for Python development, empowering users with intelligent features such as code assistance, refactoring capabilities, visual debugging support, and seamless integration with version control systems.
- **Professional (paid)** The paid version offers professional services for Python, web, and data science development. These services include code assistance, refactoring, visual debugging, integration with version control systems, remote configurations, deployment support, and assistance for popular web frameworks like Django and Flask. Additionally, it provides database support, scientific tools with Jupyter notebook integration, and tools for working with big data [70].

We used the Pycharm community, because it's free and easy to import any library you want to use, here are some of the advantages of Pycharm that we noticed while using it:

[1] www.keras.io
[2] www.numpy.org
[3] www.pandas.pydata.org
[4] www.scikit-learn.org/stable/
[5] www.www.tensorflow.org
[6] www.docs.python.org/3/library/tkinter.html

- Rich environment: PyCharm Community Edition offers a wide range of features to enhance the Python development experience. It includes code completion, refactoring tools, unit testing, debugging capabilities, and more.
- User interface: PyCharm has a user-friendly and intuitive interface, making it easy for developers, especially beginners, to navigate through their projects, write code, and access various tools and features. It offers a customizable layout and numerous themes, allowing us to personalize our development environment.
- Integrated debugging and testing: PyCharm has robust debugging capabilities that allow us to set breakpoints, step through code, inspect variables, and diagnose and fix issues.

## 2.3- ChatGPT

ChatGPT is designed to understand and generate text in a conversational manner. It can engage in discussions, answer questions, provide explanations, and offer suggestions on various topics. The model has been trained to capture context, understand nuances, and generate coherent and contextually relevant responses.

We used ChatGPT to ask him about some instructions in python for example “How to create a Button in Tkinter Python” and he  gave us the instruction. This instruction is typically provided as an example, and it needs to be adapted to suit our specific code and requirements.

## 2.4- Libraries

### 2.4.1- Keras

Keras is an API (Application Programing Interface) created with human users in mind rather than machines. It prioritizes reducing cognitive load by offering intuitive and straightforward APIs. It minimizes the need for users to perform numerous actions for typical use cases and ensures that error messages are easy to understand and act upon. Moreover, Keras places great emphasis on providing comprehensive documentation and developer guides as their top priority [71].

In our work Keras is a library used to build neural networks for deep learning models. It is designed to provide a user-friendly and efficient way of constructing them. Keras allows us to save the trained model with the '.h5' extension and import it later into another code, eliminating the need to retrain the model every time, and for example this lines from the code:

- # Create a Sequential model : `model = Sequential()`
- #Save the mdoel: `model.save("path\\deep85.h5")`

- #import the model: `model = keras.models.load_model(file_path)`

**Deep learning using keras**

According to the official website of keras[1], Keras is a high-level neural networks library, written in Python and capable of running on top of either TensorFlow or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.

Use Keras if you need a deep learning library that:

- Allows for easy and fast prototyping (through total modularity, minimalism, and extensibility).
- Supports arbitrary connectivity schemes (including multi-input and multi-output training).
- Runs seamlessly on CPU and GPU.

Here's the `Sequential` model:

```
from keras.models import Sequential

model = Sequential()
```

Stacking layers is as easy as `.add()`:

```
from keras.layers import Dense, Activation

model.add(Dense(output_dim=64, input_dim=100))
model.add(Activation("relu"))
model.add(Dense(output_dim=10))
model.add(Activation("softmax"))
```

Once your model looks good, configure its learning process with .compile():

```
model.compile(loss='categorical_crossentropy', optimizer='sgd', metrics=['accuracy'])
```

### 2.4.2- Tensorflow

TensorFlow, a Google-developed open-source library, was initially created for extensive numerical computations and not specifically for deep learning. However, its versatility led to its effective usage in deep learning applications, prompting Google to release it as an open-source platform. TensorFlow accommodates data in the form of tensors, which are multi-dimensional arrays capable of handling substantial datasets with ease. Consequently, these multi-dimensional arrays

[1] https://faroit.com/keras-docs/1.1.1/

provide a practical solution for managing vast amounts of data, making TensorFlow a valuable tool for both traditional machine learning and deep learning endeavors [72].

In our code TensorFlow is indirectly used as the backend framework for Keras can use different deep learning frameworks as its backend, including TensorFlow, Theano.

#### 2.4.3- Pandas

Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language [73].

We used pandas to read '.csv' files (read the dataset), and apply a one-hot-encoding on our data, show random samples (fifteen 15 elements ) on the GUI for the user to test the model on it. And here is examples from the code:

- #read the scv file: df= pd.read_csv("path\\output_file.csv", dtype=str)
- #encode the data: data_one_hot = pd.get_dummies(df, columns=['0', '1', '2', '3', '4', '5', '6', '7', '8'], sparse=True)

#### 2.4.3- Scikit-learn

scikit-learn is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause Berkeley Software Distribution (BSD) license. The project was started in 2007 by David Cournapeau as a Google Summer of Code project, and since then many volunteers have contributed [74].

#### 2.4.3- Tkinter

Tkinter is the de facto[1] way in Python to create GUIs and is included in all standard Python Distributions. In fact, it's the only framework built into the Python standard library.

This Python framework provides an interface to the Tk toolkit and works as a thin object-oriented layer on top of Tk. The Tk toolkit is a cross-platform collection of 'graphical control elements', also known as widgets, for building application interfaces [75].

## 3- Corpus

We used the OntoNotes 5.0 annotated corpus and its preprocessing in [4].

### 3.1- OntoNotes 5.0

[1] In the context of programming and software development, "de facto" refers to something that has become widely accepted or adopted as the standard, even if there is no official or mandated standard.

Funded by the US Department of Defense, the collaborative OntoNotes project involving the following institutions marks its culmination in OntoNotes Release 5.0:

- BBN Technologies.
- The University of Colorado.
- The University of Pennsylvania.
- The University of Southern California's Information Sciences Institute.

This project aimed to enrich a vast collection of texts from diverse genres (such as news, conversational telephone speech, weblogs, usenet newsgroups, broadcast, and talk shows) in three languages (English, Chinese, and Arabic) with essential structural information (including syntax and predicate argument structure) and surface-level semantics (such as word sense connected to an ontology and coreference).

OntoNotes Release 5.0 incorporates the contents of its earlier versions, namely OntoNotes Release 1.0 LDC2007T21, OntoNotes Release 2.0 LDC2008T04, OntoNotes Release 3.0 LDC2009T24, and OntoNotes Release 4.0 LDC2011T03. In addition, it includes new source data and/or annotations for various categories such as newswire (News), broadcast news (BN), broadcast conversation (BC), telephone conversation (Tele), and web data (Web) in both English and Chinese. Moreover, it encompasses newswire data in Arabic and English pivot text from the Old Testament and New Testament. This comprehensive release encompasses a total of 2.9 million words, and the corresponding word counts are provided in the table [4.1] [76].

Table 4.1: OntoNotes Release 5.0: Word Count Summary by Language and Source [76]

| | Arabic | English | Chinese |
|---|---|---|---|
| **News** | 300K | 625k | 250k |
| **BN** | n/a | 200k | 250k |
| **BC** | n/a | 200K | 150k |
| **Web** | n/a | 300K | 150k |
| **Tele** | n/a | 120K | 100k |
| **Pivot** | n/a | n/a | 300 |

## 3.2- CoNLL

During the CoNLL-2012 conference, the data utilized for English, Chinese, and Arabic is derived from OntoNotes 5.0 and employed in a coreference task. The organizers utilize a specific algorithm to generate the development, training, and test sets.

Within the Arabic annotations folder in OntoNotes 5.0, there are seven (07) extensions that correspond to different annotation levels, ***namely conef, lemma, name, conf, parse, prop, sens,*** and **source**. These extensions are distributed across six (06) folders, as illustrated in Figure [4.2]. [4.3] demonstrates the annotation of a sentence from an extension file (.conf) [4].

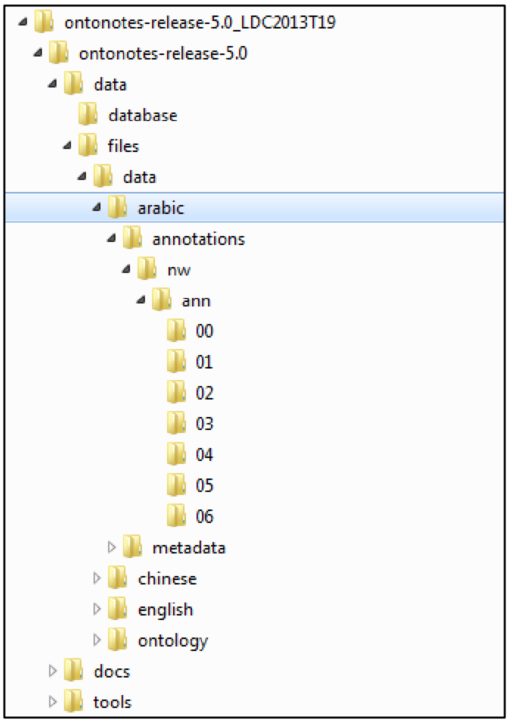



Figure 4.2: Arabic folder in OntoNotes 5.0 [4]

```
26  ما
27  يُحيط
     * prop:  OaHAT.01
       v           * -> 27:0, يُحيط
       ARG1        * -> 26:1, ما
                   * -> 27:1, بِ- -نا الآنَ مِن أوضاع إقليميّةً وَ- -خارجيّةٍ *T*-3 يُحيط
                   * -> 28:0, *T*-3 -> 26:1, ما
                   * -> 32:1, مِن أوضاع إقليميّةً وَ- -خارجيّةٍ
       ARG2        * -> 29:1, بِ- -نا
       ARGM-TMP    * -> 31:1, الآنَ
28  *T*-3
29  بِ-
30  -نا
     coref: IDENT        75042 30-30   نا
31  الآنَ
32  مِن
33  أوضاع
34  إقليميّةً
35  وَ-
36  -خارجيّةٍ
37  يُحَتِّمُ
     * prop:  Hat~am.01
       v           * -> 37:0, يُحَتِّمُ
       ARG0        * -> 26:2, بِ- -نا الآنَ مِن أوضاع إقليميّةً وَ- -خارجيّةٍ *T*-3 ما يُحيط
                   * -> 27:2, بِ- -نا الآنَ مِن أوضاع إقليميّةً وَ- -خارجيّةٍ *T*-3 يُحيط
                   * -> 38:0, *T*-2 -> 26:2, -بِ- -نا الآنَ مِن أوضاع إقليميّةً وَ *T*-3 ما يُحيطُ
                                                       -خارجيّةٍ
       ARG1        * -> 39:1, عَلَي- -نا
       ARG1        * -> 41:3, *T*-4 وَعياً في التَّصَرُّفِ , وَ- -إدراكاً في إبعادِ كُلِّ ما نَقُومُ * بِ- -هِ
                              مِن خَطَواتٍ
       ARGM-CAU    * -> 59:2, لِ- -ذا عَلَي- -نا أن نَكُونَ * مُنتَبِهِينَ جِدّاً , وَ- -يَقِظِينَ , فَ
                              -عَلَي- -نا في إستِمرارِ , وَ *T*-5 المُناخاتُ غَيرُ الصافِيَةِ الَّتي تَهُبُّ-
                              مِن كُلِّ حدبٍ وَ- -صَوبٍ وَ- -في كُلِّ 6- * *T*-6 الضُّغُوطُ الَّتي تُمارَسُ-
                              ال{تِّجاهاتِ السِياسِيَّةِ وَ- -ال{قتِصادِيَّةِ وَ- -ال{جتِماعِيَّةِ يَجِبُ أن
                              -نا إلَى المَزِيدِ مِن التَّعاوُنِ وَ- *T*-8 -حافِزاً 0 يَندَفِعُ *T*-7 تَكُونَ
                              شَرّاً لِ- -ه`ذا *T*-9 التَّوافُقِ لِ- -تَفوِيتِ الفُرَصِ أمامَ كُلِّ مَن يُضمِرُ-
                              الأَثمانَ باطِنَةً , وَ- 0 -لا يَزالُ *T*-10 الوَطَنِ الَّذِي دَفَعَ شَعبُ- -هُ
                              *T*-11 يُكابِدُ * آثارَ الإرهابِ الصَهيُونيِّ الَّذِي لَم يُطاوِلْ *T*-13
                              الفِلَسطِينِيِّينَ وَحدَ- -هُم , بَل لُبنانَ وَ- -سُوريا وَ- -كُلَّ شُعُوبِ
                              المِنطَقَةِ
```

Figure 4.3: Labeling of Arabic sentence [4]

This extension is in a specialized form, similar to a table with ***n* rows** and ***n* columns**. Each row represents a constituent of a sentence, and each column represents a piece of information such as files, word number, root, part of speech, etc [4].

In the table [4.2] we show Format of the .conll file used in the shared task.

Table 4.2: Format of the .conll file used in the shared task [77]

| Column | Type | Description |
|---|---|---|
| 1 | Documant ID | This is a variation on the document filename |
| 2 | Part number | Some files are divided into multiple parts numbered as 000, 001, 002, ... etc |
| 3 | Word number | This is the word index in the sentence |
| 4 | Word | The word itself |
| 5 | Part of Speech | Part of Speech of the word |
| 6 | Parse bit | This is the bracketed structure broken before the first open parenthesis in the parse, and the word/part-of-speech leaf replaced with a *. The full parse can be created by substituting the asterisk with the ([pos] [word]) string (or leaf) and concatenating the items in the rows of that column. |
| 7 | Lemma | The predicate/sense lemma is mentioned for the rows for which we have semantic role or word sense information. All other rows are marked with a |
| 8 | Predicate Frameset ID | This is the PropBank frameset ID of the predicate in Column 7. |
| 9 | Word sense | This is the word sense of the word in Column 4 |
| 10 | Speaker/Author | This is the speaker or author name where available. Mostly in Broadcast Conversation and Weblog data. |
| 11 | Named Entities | These columns identifies the spans representing various named entities |
| 12:N | Predicate Arguments | There is one column each of predicate argument structure information for the predicate mentioned in Column 7. |
| N | Coreference | Coreference chain information encoded in a parenthesis structure. |

In Figure [4.4] displays an example of a **conll** training file [4]. According to the same author, these data possess various strengths, such as:

- Validated by the NLP community.
- A significant number of annotated semantic roles (the largest) .
- Developed by prominent American institutions.

The table [4.3] shows the arguments of the database.

Table 4.3: Arguments of the database

| Simple Arguments | ARG-M | C-ARG and R-ARG |
| --- | --- | --- |
| ARG0 | ARGM-ADV | C-ARG0 |
| ARG3 | ARGM-PRP | C-ARGM-PRP |
| ARG1 | ARGM-LOC | C-ARG1 |
| ARG4 | ARGM-TMP | C-ARGM-TMP |
| ARG2 | ARGM-MNR | C-ARG2 |
| | ARGM-COM | R-ARG0 |
| | ARGM-NEG | C-ARGM-ADV |
| | ARGM-CAU | R-ARG1 |
| | ARGM-GOL | C-ARGM-LOC |
| | ARGM-EXT | R-ARG2 |
| | | ARGMADV |
| | | C-ARG2 |
| | | R-ARGM-TMP ARGMADV |
| | | C-ARG1 |
| | | R-ARGM-LO |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 : N | | N+1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| nw/ann/02/ann_0290 | 0 | 0 | [WORD] | PSEUDO_VERB | (TOP(S(VP* | [LEMMA] | - | - | - | * | * | * | - |
| nw/ann/02/ann_0290 | 0 | 1 | [WORD] | PRON_3MS | (NP*) | [LEMMA] | - | - | - | * | (ARG0*) | (ARG0*) | (6) |
| nw/ann/02/ann_0290 | 0 | 2 | [WORD] | NEG_PART | (S(VP(PRT*) | [LEMMA] | - | - | - | * | * | * | - |
| nw/ann/02/ann_0290 | 0 | 3 | [WORD] | IV3MS+IV+IVSUFF_MOOD:J | * | [LEMMA] | 01 | - | - | * | (V*) | * | - |
| nw/ann/02/ann_0290 | 0 | 4 | [WORD] | PREP | (PP* | [LEMMA] | - | - | - | * | (ARG1* | * | - |
| nw/ann/02/ann_0290 | 0 | 5 | [WORD] | NOUN+CASE_DEF_GEN | (NP* | [LEMMA] | - | - | - | * | * | * | - |
| nw/ann/02/ann_0290 | 0 | 6 | [WORD] | DET+NOUN+NSUFF_FEM_PL+CASE_DEF_GEN | (NP*))) | [LEMMA] | - | - | - | * | *) | * | - |
| nw/ann/02/ann_0290 | 0 | 7 | [WORD] | DET+NOUN+CASE_DEF_ACC | (NP* | [LEMMA] | - | 4 | - | * | (ARGM-TMP* | * | - |
| nw/ann/02/ann_0290 | 0 | 8 | [WORD] | CONJ | * | [LEMMA] | - | - | - | * | * | * | - |
| nw/ann/02/ann_0290 | 0 | 9 | [WORD] | NOUN+CASE_INDEF_ACC | *) | [LEMMA] | - | - | - | * | *) | * | - |
| nw/ann/02/ann_0290 | 0 | 10 | [WORD] | PREP | (PP* | [LEMMA] | - | - | - | * | (ARGM-MNR* | * | - |
| nw/ann/02/ann_0290 | 0 | 11 | [WORD] | SUB_CONJ | (SBAR* | [LEMMA] | - | - | - | * | * | * | - |
| nw/ann/02/ann_0290 | 0 | 12 | [WORD] | PV+PVSUFF_SUBJ:3MS | (S(VP* | [LEMMA] | 01 | - | - | * | * | (V*) | - |
| nw/ann/02/ann_0290 | 0 | 13 | [WORD] | NOUN+CASE_INDEF_ACC | (NP*)))))))) | [LEMMA] | - | - | - | * | *) | (ARGM-TMP*) | - |
| nw/ann/02/ann_0290 | 0 | 14 | [WORD] | PUNC | *)) | [LEMMA] | - | - | - | * | * | * | - |

Figure 4.4: Example of a file in conll format [4]

And the database contains ten (10) columns described in the table [4.4]:

Table 4.4: Columns names of the database

| Column | Type |
|---|---|
| 1 | Attribute (verb) |
| 2 | Lemma attribute frameset |
| 3 | ID frameset |
| 4 | first constituent |
| 5 | Lemma first constituent |
| 6 | Position du first constituant par rapport à |
| 7 | Place of the 1st component in relation to |
| 8 | parts of speech |
| 9 | Analysis |
| 10 | Argument |

## 3.3- Statistics on the database

We wrote a Python program that reads the database and gives us statistics on the distribution of classes as it is shown in the histogram figure [4.5.a][4.5.b].

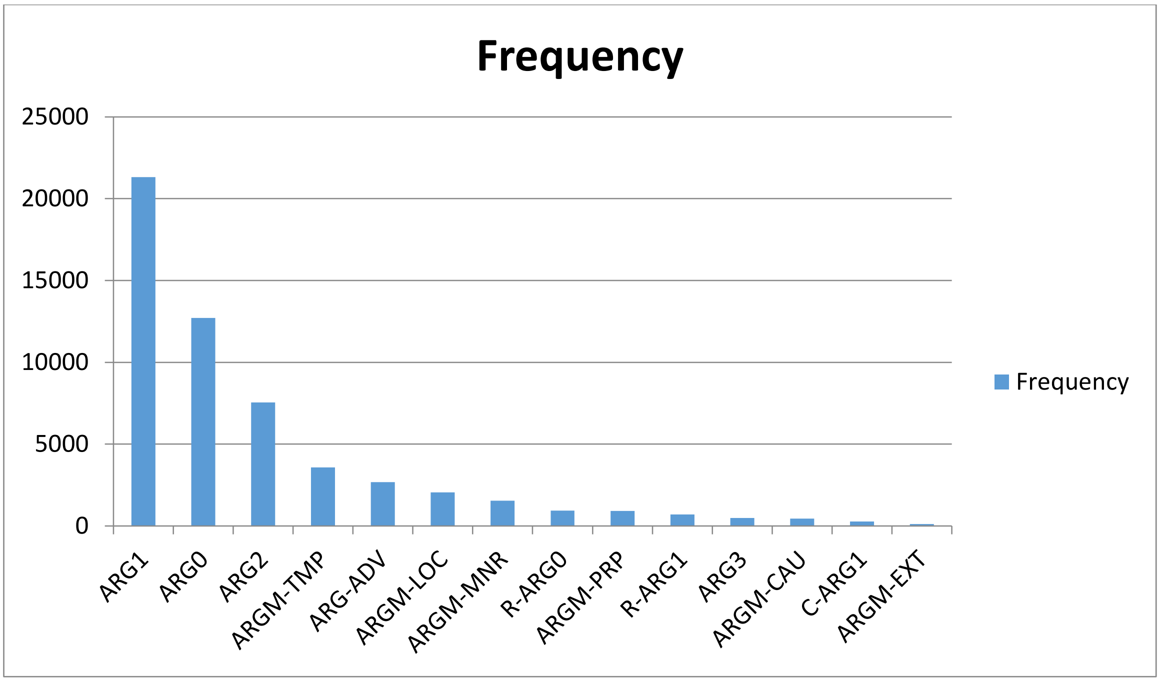



Figure 4.5.a: A histogram show distribution of classes in the database

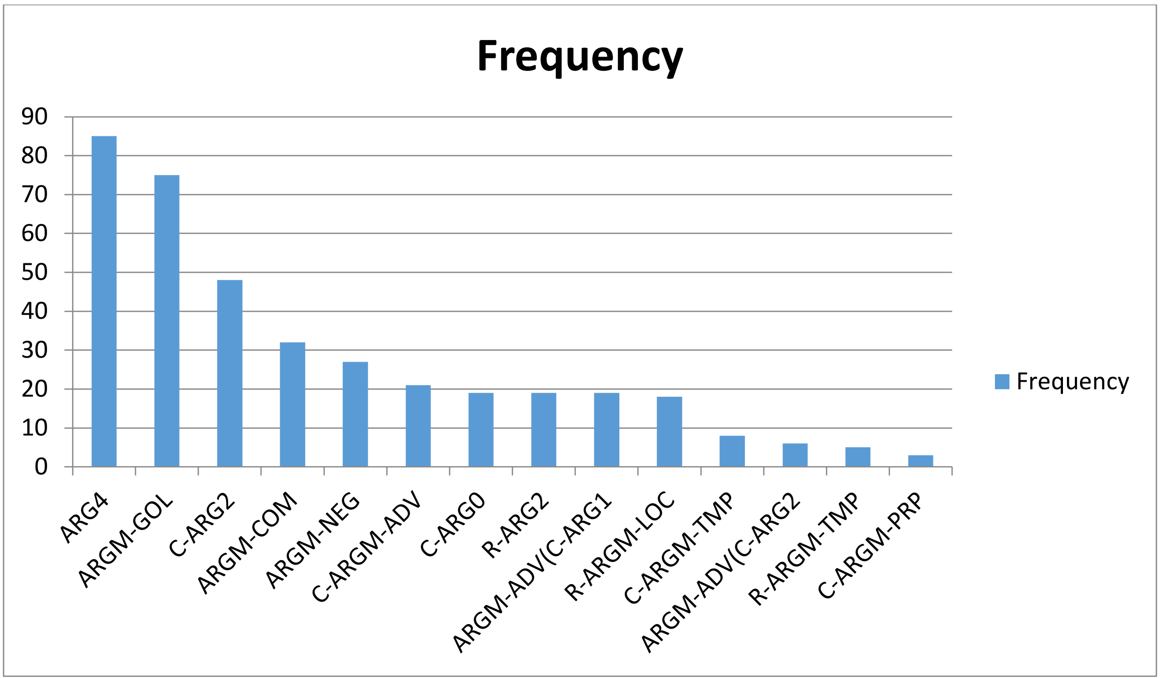



Figure 4.5.b: A histogram show distribution of classes in the database

In addition to this table [4.5] that represents statistics from our data set:

Table 4.5: The distribution of classes in the database

| Stats | Value | Count | % |
|---|---|---|---|
| **Rows** | / | 55715 | 100% |
| **Columns** | / | 10 | / |
| **Class 1** | ARG1 | 21318 | 38.25% |
| **Class 2** | ARG0 | 12719 | 22.83% |
| **Class 3** | ARG2 | 7541 | 13.53% |
| **Class 4** | ARGM-TMP | 3584 | 6.43% |
| **Class 5** | ARG-ADV | 2681 | 4.81 |
| **Class 6** | ARGM-LOC | 2045 | 3.67% |
| **Class 7** | ARGM-MNR | 1541 | 2.77% |
| **Class 8** | R-ARG0 | 938 | 1.68% |
| **Class 9** | ARGM-PRP | 924 | 1.66% |
| **Class 10** | R-ARG1 | 706 | 1.27% |
| **Class 11** | ARG3 | 494 | 0.89% |
| **Class 12** | ARGM-CAU | 451 | 0.81% |
| **Class 13** | C-ARG1 | 266 | 0.48% |
| **Class 14** | ARGM-EXT | 120 | 0.22% |
| **Class 15** | ARG4 | 85 | 0.15% |
| **Class 16** | ARGM-GOL | 75 | 0.13% |

| | | | |
|---|---|---|---|
| **Class 17** | C-ARG2 | 48 | 0.09% |
| **Class 18** | ARGM-COM | 32 | 0.06% |
| **Class 19** | ARGM-NEG | 27 | 0.05% |
| **Class 20** | C-ARGM-ADV | 21 | 0.04% |
| **Class 21** | C-ARG0 | 19 | 0.03% |
| **Class 22** | R-ARG2 | 19 | 0.03% |
| **Class 23** | ARGM-ADV(C-ARG1 | 19 | 0.03% |
| **Class 24** | R-ARGM-LOC | 18 | 0.03% |
| **Class 25** | C-ARGM-TMP | 8 | 0.01% |
| **Class 26** | ARGM-ADV(C-ARG2 | 6 | 0.01% |
| **Class 27** | R-ARGM-TMP | 5 | 0.01% |
| **Class 28** | C-ARGM-PRP | 3 | 0.01% |

## 4- Experiments and building the model

Based on the result of [54], we have selected deep learning as a technology to build our model. We tested several neural networks and the table [4.6] shows the results for each experimentation of neural network:

Table 4.6: Experiments of building a neural network for the model

| Experiment | Layer 1 | | Layer 2 | | Layer 3 | | Layer 4 | | Layer 5 | | Result (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Neural Nbr | Activation | Neural Nbr | Activation | Neural Nbr | Activation | Neural Nbr | Activation | Neural Nbr | Activation | / |
| **1** | 64 | tanh | 128 | tanh | 256 | tanh | 512 | tanh | / | / | **61.84** |
| **2** | 1024 | LeakyRelu | 1024 | LeakyRelu | 1024 | LeakyRelu | 1024 | LeakyRelu | / | / | **79.07** |
| **3** | 512 | relu | 1024 | relu | 512 | relu | 4 | relu | / | / | **78.24** |
| **4** | 1500 | relu | 1000 | relu | 990 | relu | 800 | relu | 29 | softmax | **81.80** |

In all experiments, the input layer has 9 neurons, and the output layer has 28 layers.

All layers were of type “Dense” And it is full connective layer, and we used “adam” as optimizer and “sparse_categorical_crossentropy” as loss functions.

In our experimentation Table [4.6], the rows indicate the experiment number, and the columns represent the layers of the neural network. Each layer has two attributes:

- Neural Nbr: the number of neurons on this layer .
- Activation: the Activation function that used on this layer.

## 4.1 First experiment

In the first experiment, we used four (04) layers:

- ✓ In the first one, there were 64 neurons.
- ✓ 128 neurons in the second.
- ✓ 256 in the third.
- ✓ 512 neurons in the last one.
- ✓ For the activation function, we used "tanh" (Hyperbolic Tangent).

This architecture gave us 61.84% of accuracy.

## 4.2 Second experiment

In the second experiment, we used four (04) layers, there were 1024 neurons on each one. “LeakyRelu” as an activation function, this architecture gave us 79.07% of accuracy.

## 4.3 Third experiment

In the third experiment, we used four (04) layers:

- ✓ In the first one, there are 512 neurons.
- ✓ 1024 neurons in the second.
- ✓ 512 in the third, and 4 neurons in the last one.
- ✓ “relu” as an activation function.

This architecture gave us 78.24% of accuracy.

## 4.4 Fourth experiment

In the fourth experiment, we used five (05) layers:

- In the first one, there are 1500 neurons.
- 1000 neurons in the second.

- 990 neurons in the third.
- 800 neurons in the fourth layer “relu” as an activation function.
- 29 neurons in the last one, followed by “softMax” as an activation function.

This architecture gave us 81.80% of accuracy.

### 4.5 Overfitting

After the fifth experiment, It became impossible to increase the ratio by modifying the number and structure of neurons (Overfitting). Therefore, we conducted a statistical analysis of the results, displaying the error percentage for each category, in order to identify the categories that had a negative impact on the results. We performed this analysis three times and obtained the following results As shown in the tables [4.7][4.8][4.9]:

Table 4.7: The first try of showing the percentage of failed cases for each class

| Class | Failed | Class | Failed | Class | Failed | Class | Failed |
|---|---|---|---|---|---|---|---|
| **ARG0:** | 6.01% | **ARGM-NEG:** | 33.33% | **ARGM-MNR:** | **52.63%** | **ARGM-COM** | 100.00% |
| **ARG1:** | 12.23% | **ARG3:** | 34.04% | **C-ARG1:** | **54.17%** | **ARGM-ADV** | 100.00% |
| **ARGM-TMP:** | 22.99% | **ARGM-ADV:** | 36.73% | **ARGM-EXT:** | **63.16%** | **R-ARG2** | 100.00% |
| **ARG2:** | 24.36% | **ARGM-PRP:** | 42.50% | **C-ARG2:** | **71.43%** | **R-ARGM-LOC** | 100.00% |
| **R-ARG0:** | 25.00% | **ARGM-LOC:** | 47.25% | **C-ARG0:** | **100.00%** | | |
| **R-ARG1:** | 26.32% | **ARGM-CAU:** | 49.02% | **C-ARGM-ADV** | **100.00%** | | |
| **ARG4:** | 33.33% | ARGM-GOL: | 50.00% | C-ARGM-LOC | 100.00% | | |

Table 4.8: The second try of showing the percentage of failed cases for each class

| Class | Failed | Class | Failed | Class | Failed | Class | Failed |
|---|---|---|---|---|---|---|---|
| **ARG0:** | 5.55% | **ARGM-NEG:** | 33.33% | **ARGM-MNR:** | 56.73% | **ARGM-COM** | 100.00% |
| **ARG1:** | 10.89% | **ARG3:** | 36.17% | **C-ARG1:** | 62.50% | **ARGM-ADV(C-ARG1** | 100.00% |
| **ARGM-TMP:** | 24.18% | **ARGM-ADV:** | 35.92% | **ARGM-EXT:** | 57.89% | **R-ARG2** | 100.00% |
| **ARG2:** | 24.90% | **ARGM-PRP:** | 37.50% | **C-ARG2:** | 85.71% | **R-ARGM-LOC** | 100.00% |
| **R-ARG0:** | 27.27%% | **ARGM-LOC:** | 35.71% | **C-ARG0:** | 100.00% | | |
| **R-ARG1:** | 34.21% | **ARGM-CAU:** | 68.63% | **C-ARGM-ADV** | 100.00% | | |
| **ARG4:** | 33.33% | **ARGM-GOL:** | 50.00% | **C-ARGM-LOC** | 100.00% | | |

Table 4.9: The third try of showing the percentage of failed cases for each class

| Class | Failed | Class | Failed | Class | Failed | Class | Failed |
|---|---|---|---|---|---|---|---|
| **ARG0:** | 9.95% | **ARGM-NEG:** | 0.00% | **ARGM-MNR:** | 66.08% | **ARGM-COM** | 100.00% |
| **ARG1:** | 7.80% | **ARG3:** | 38.30% | **C-ARG1:** | 75.00% | **ARGM-ADV(C-ARG1** | 100.00% |
| **ARGM-TMP:** | 24.48% | **ARGM-ADV:** | 33.47% | **ARGM-EXT:** | 68.42% | **R-ARG2** | 100.00% |
| **ARG2:** | 27.04% | **ARGM-PRP:** | 38.75% | **C-ARG2:** | 85.71% | **R-ARGM-LOC** | 100.00% |
| **R-ARG0:** | 29.55% | **ARGM-LOC:** | 45.60% | **C-ARG0:** | 100.00% | | |
| **R-ARG1:** | 27.63% | **ARGM-CAU:** | 76.47% | **C-ARGM-ADV** | 100.00% | | |
| **ARG4:** | 33.33% | **ARGM-GOL:** | 33.33% | **C-ARGM-LOC** | 100.00% | | |

the tables [4.7][4.8][4.9] show the percentage of failed predictions for each class. We used these tables to identify the classes that have a negative impact on the model.

By examining these tables, we identified classes that exhibit a 100% failure rate, Which greatly affects the performance of the model. Table [4.5] reveals that these classes lack data for learning, which leads to erroneous model predictions.

To address this issue, we tried to consolidate classes with high failure rates into a single class called "ARG-Mix". Consequently, we retained only the classes (ARG1, ARG0, ARG-Mix, ARG2, ARGM-TMP) in a new database named "new_db.csv".

The table [4.10] shows the updated statistics of the "new_db.csv" database.

Table 4.10: The new distribution of classes on the database (After adding ARG-Mix)

| Stats | Value | Count | % |
|---|---|---|---|
| **Columns** | / | 10 | / |
| **Rows** | / | 55715 | 100% |
| **Class 1** | ARG1 | 21318 | 38.26% |
| **Class 2** | ARG0 | 12719 | 22.83% |
| **Class 3** | ARG-Mix | 10553 | 18.94% |
| **Class 4** | ARG2 | 7541 | 13.53% |

| | | | |
|---|---|---|---|
| **Class 5** | ARGM-TMP | 3584 | 6.43% |

We had five (05) classes. The first class (ARG1) got a percentage of 38.26%, while the second class (ARG0) got a percentage of 22.83%, while the other classes (ARG-Mix, ARG2 and ARGM-TMP) got a percentage of 18.94% and 13.53%. and 6.43%, respectively. Below (Fig. 4.6) we show the content of the table.

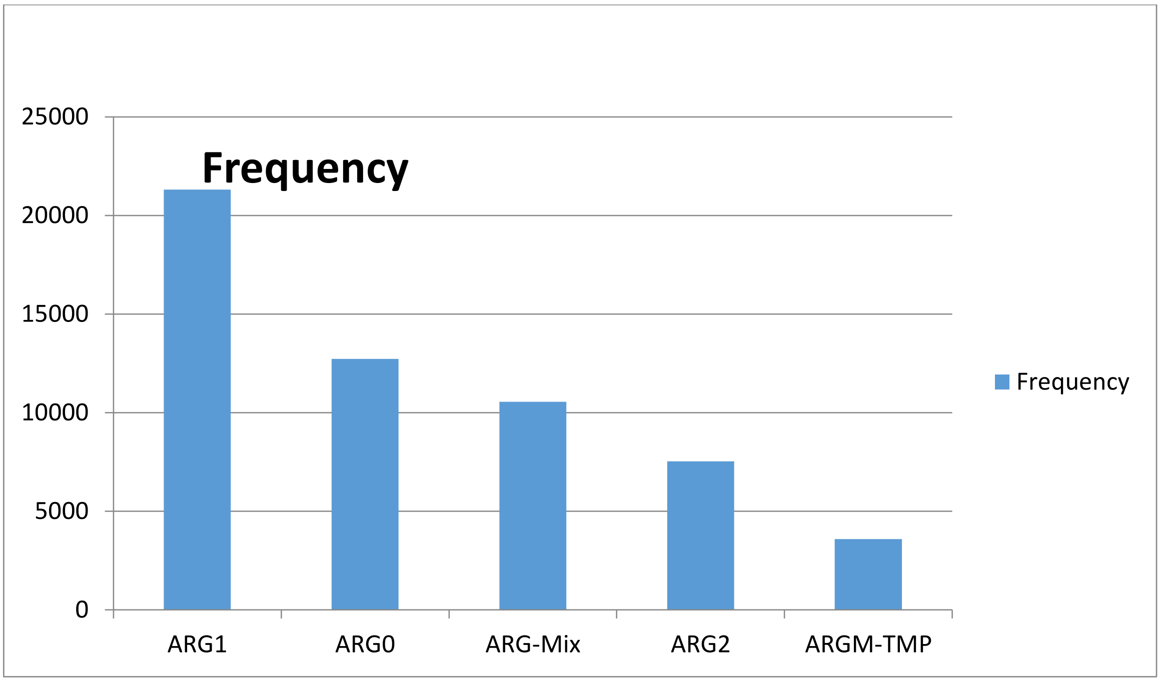



Figure 4.6: A histogram show distribution of the new classes in the database

After training and testing the same model on the "new_db.csv," we got accuracy of 85.24%. The result increased from 81.80% to 85.24%, and this improved accuracy will assist annotators to identify the label of each sentence.

The 3.44% increase in accuracy, gives us nearly two thousand cases from the database that should reduce the amount of manual work for researchers in SRL. The failure rate for each class is provided in the table [4.11].

Table 4.11: The failed of each class after adding ARG-Mix

| Class | Fail |
|---|---|
| ARG0 | 6.32% |
| ARG1 | 12.97% |
| ARG-Mix | 20.18% |
| ARG2 | 20.88% |
| ARGM-TMP | 28.66% |

# 5 - AI models hybridization

We extracted the failed cases (14.76%) from the previous model and we saved them in a CSV file for testing on another model.

The machine learning model architecture was like that:

- Input layer: 9 neurons
- Hidden layer 1: 1000 neurons
- Hidden layer 2: 993 neurons
- Hidden layer 3: 250 neurons
- Output layer : 5 neurons

This new model based on machine learning, it gave us 23% accuracy of the failed cases. That 23% represents 3.2% of all dataset (Data_Big_New_v11.0.csv), if we combine the predictions of both models, we achieve an overall accuracy of 88.44%.

# 6- A tool for semantic roles labeling

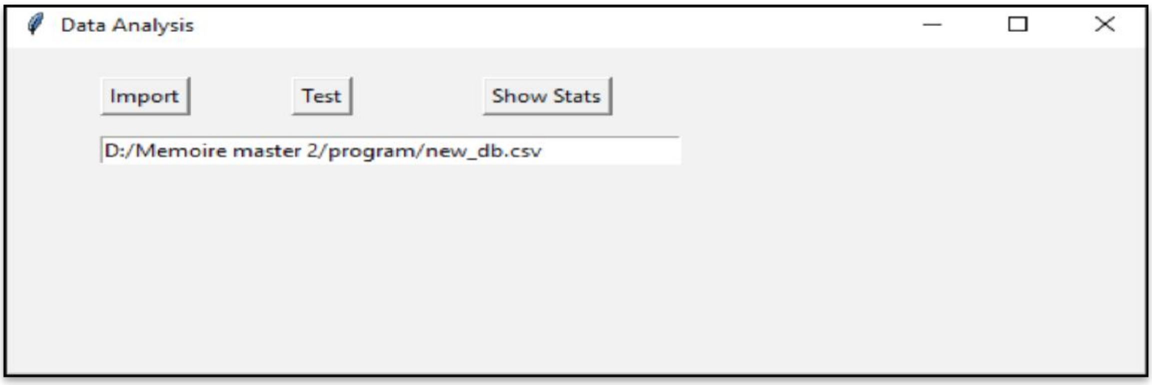
## 6.1- Main window

Using the Tkinter[1] library in Python, we have developed a GUI application that utilizes the "deep85.h5" model to predict the labeling of semantic roles for a test set.

- The main window of our application figure[4.7] displays a text field and a button to import the database.
- Clicking the "Stats" button opens a separate window displaying statistical information.

1 https://docs.python.org/3/library/tkinter.html

- The "Test" button opens a window to perform model testing on the imported dataset.




Here is a screenshot of "main.py."

Figure 4.7: The main window of our application

## 6.2- Stats window

The 'Stats' window figure [4.8] is a Tkinter GUI that enables the user to consult the database, view class statistics, and analyze data distribution.

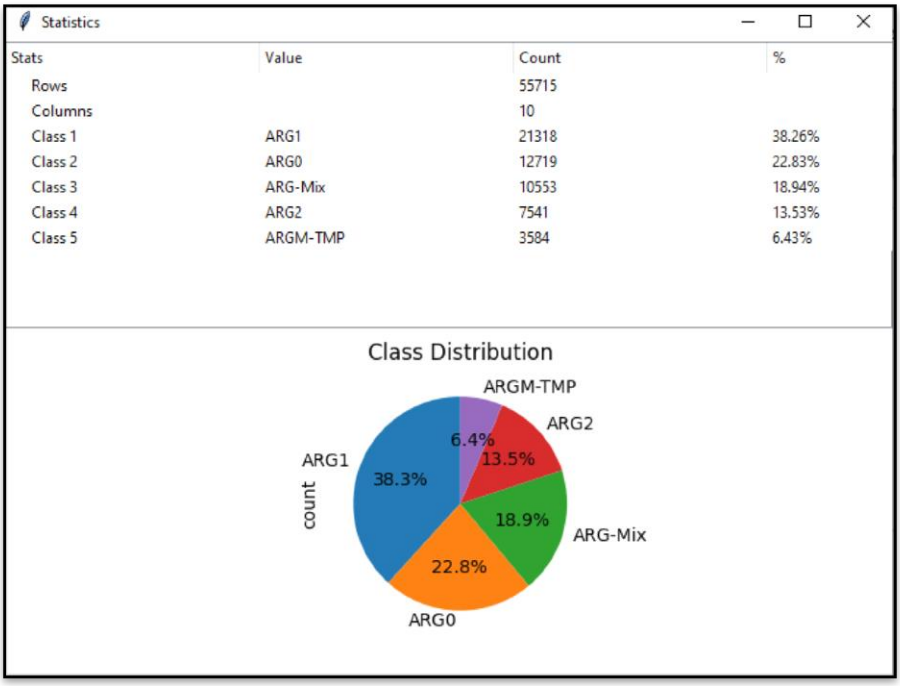



Figure 4.8: Stats window on the database

## 6.3- Test window

A window is provided to allow users to import the SRL model "deep85.h5" and “MLmodel.pkl” to test it on the imported database.

This window show in figure [4.9] displays a random selection of 15 rows from the dataset, and the user can choose one of these rows and click "Test sample". This action takes the selected row, passes it to the model and displays the prediction in a messagebox.

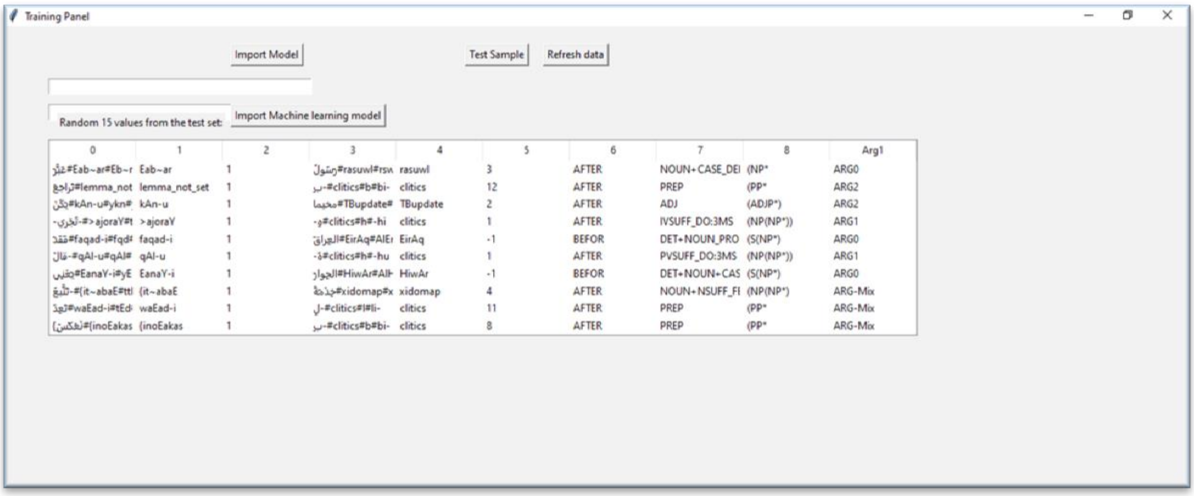



Figure 4.9: Training window on our SRL application

## 6.4- The prediction

After clicking the "Test Sample" button, the model will generate a prediction for the selected row and display it in a message box As shown in figure [4.10].

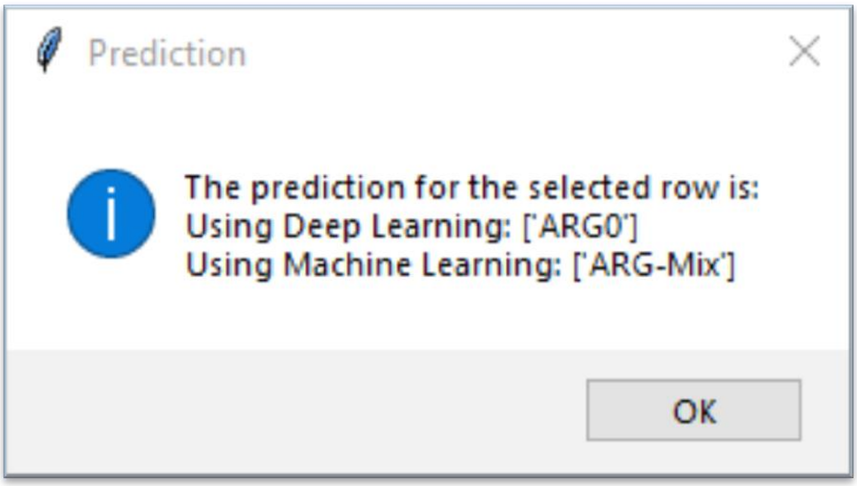


Figure 4.10: The message box that shows the result of prediction

## 7 – Conclusion

In this chapter we have addressed our research work by reviewing implimentation, experiments and results.

First, we trained a deep learning model, and after conducting four (04) experiments, we achieved an accuracy of 81.80%. In these experiments, we explored different attributes such as activation functions and the number of neurons. According to the statistics, we identified certain classes that not had an enough cases for training. To address this, we combined these classes into the ARG-Mix class. This step resulted in an increased accuracy of 85.24%.

Afterward, we analyzed the failure cases of the deep learning model and applied a machine learning model to address them. The machine learning model achieved an accuracy of 23%, which is an improvement of 3%. As a result, the overall accuracy increased from 85.24% to 88.24%.

We integrated these two models into our application, which enables annotators to predict semantic roles from the OntoNotes 5.0.

# Conclusion And Future Work

# Conclusion and Future Work

In this thesis, we have tackled a significant topic in the field of natural language processing. Our focus was on developing an annotation support tool for the Arabic semantic roles, using machine learning methods.

When we talk about the Arabic language in the field of NLP, we will find very few works compared to other languages such as English, and this is one of the difficulties we encountered with the lack of resources, which explains why researchers avoid working on this language. Our work had two main objectives, improving results over previous research and developing an annotation support tool.

When it comes to the Arabic language in the field of NLP, there is a noticeable scarcity of research compared to other languages like English. This scarcity is primarily attributed to the lack of resources, which presents a significant challenge and discourages researchers from working on this language. Our work aimed to address two main objectives: enhancing the results compared to previous research and developing an annotation support tool.

Our contribution to improve the results focused on the use of deep learning, because it gave the best results in the contribution that preceded us. We achieve an accuracy of 85%. However, we didn't stop there, we combined it with machine learning, resulting is improved and and got a result of 88%.

The tool we have developed harnesses the power of deep learning and machine learning. With this tool, users can select any unannotated sentence from the database and with a simple click, they receive two suggested label generated by the two models. This enables the user to quickly and efficiently annotate the sentence with semantic roles.

In the end, we hope that our tool will help annotators to develop conll database more, in order to improve the Arabic language in the field of natural language processing, as it facilitates their work, reduces the need for Arabic linguists, and reduces the effort and time required for this work.

The main difficulty we faced in this work was the scarcity of sentences in some features, to the extent that they were not sufficient for learn and training the models. We mentioned this problem in Chapter Four and solve it by augmenting the limited features. Therefore, we expect future works in this field to focus on increasing the number of sentences for the mentioned features in order to enhance the accuracy of the models.

Our tool holds the potential to assist annotators in expanding the database. By streamlining their workflow, reducing the dependency on Arabic linguists, and minimizing the effort and time required for annotation.

Throughout our work, we encountered a significant challenge related to the scarcity of sentences for certain features. These limitations hindered the learning and training of our models. In Chapter Four, we addressed this issue by employing augmentation techniques to supplement the limited feature set. Moving forward, we recommend research in this field focuses on increasing the availability of sentences to further enhance the accuracy of the models.

# References

# References


[1] K. R. Chowdhary, “Natural Language Processing,” in *Fundamentals of Artificial Intelligence*, K. R. Chowdhary, Ed., New Delhi: Springer India, 2020, pp. 603–649. doi: 10.1007/978-81-322-3972-7_19.

[2] M. Diab, A. Moschitti, and D. Pighin, “CUNIT: a semantic role labeling system for modern standard Arabic,” in *Proceedings of the 4th International Workshop on Semantic Evaluations - SemEval ’07*, Prague, Czech Republic: Association for Computational Linguistics, 2007, pp. 133–136. doi: 10.3115/1621474.1621500.

[3] H. Meguehout, T. Bouhadada, and M. T. Laskri, “Semantic Role Labeling for Arabic Language Using Case-based Reasoning Approach,” *International Journal of Speech Technology*, vol. 20, no. 2, Art. no. 2, Juin 2017, doi: 10.1007/s10772-017-9412-6.

[4] H. MEGUEHOUT, “Un raisonnement à partir de cas (RàPC) pour la traduction automatique du langage naturel (de l’Arabe vers le Français),” Thèse de Doctorat, BADJI MOKHTAR, Annaba, Algérie, 2009.

[5] T. O’Hara and J. Wiebe, “Exploiting Semantic Role Resources for Preposition Disambiguation,” *Computational Linguistics*, vol. 35, no. 2, pp. 151–184, Jun. 2009, doi: 10.1162/coli.06-79-prep15.

[6] I. N. Kardana, I. M. B. Wrihatnala, and M. S. Satyawati, “Category Of Complement And Semantic Role Of Single Argument In Balinese Syntactic Constructions,” *RETORIKA: Jurnal Ilmu Bahasa*, vol. 2, no. 2, Art. no. 2, 2016, doi: 10.22225/jr.2.2.67.384-393.

[7] M. D. Sidabutar and Z. Zakrimal, “Semantic Roles in Joko Widodo Re-Elected as President of BBC Online News,” *Linguistic, English Education and Art (LEEA) Journal*, pp. 56–65, Jul. 24, 2020.

[8] J. Zhou and W. Xu, “End-to-end learning of semantic role labeling using recurrent neural networks,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 1127–1137. doi: 10.3115/v1/P15-1109.

[9] R. Johansson and P. Nugues, “Dependency-based Semantic Role Labeling of PropBank,” in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Honolulu, Hawaii: Association for Computational Linguistics, Oct. 2008, pp. 69–78. Accessed: May 17, 2023. [Online]. Available: https://aclanthology.org/D08-1008

[10] “[PDF] Frame semantics for text understanding | Semantic Scholar.” https://www.semanticscholar.org/paper/Frame-semantics-for-text-understanding-Fillmore-Baker/72cc80486744320d5b9a2ea75e3fb1cde4ca669c (accessed May 17, 2023).

[11] Z. Tan, M. Wang, J. Xie, Y. Chen, and X. Shi, “Deep Semantic Role Labeling With Self-Attention,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Art. no. 1, Apr. 2018, doi: 10.1609/aaai.v32i1.11928.

[12] F. Kong, H. T. Ng, and G. Zhou, “A Constituent-Based Approach to Argument Labeling with Joint Inference in Discourse Parsing,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 68–77. doi: 10.3115/v1/D14-1008.

[13] L. Màrquez, X. Carreras, K. C. Litkowski, and S. Stevenson, “Special Issue Introduction: Semantic Role Labeling: An Introduction to the Special Issue,” *Computational Linguistics*, vol. 34, no. 2, pp. 145–159, 2008, doi: 10.1162/coli.2008.34.2.145.

[14] E. Loper, S. Yi, and M. Palmer, “Combining Lexical Resources: Mapping Between PropBank and VerbNet”.

[15] K. K. Schuler, “Verbnet: A Broad-coverage, Comprehensive Verb Lexicon,” Thèse de Doctorat, Université de Pennsylvania, États-Unis, 2005. [Online]. Available: https://books.google.dz/books?id=VMY-OAAACAAJ

[16] B. Levin, *English Verb Classes and Alternations: A Preliminary Investigation*. Université de Chicago, États-Unis: University of Chicago Press, 1993. [Online]. Available: https://books.google.dz/books?id=6wIZWOrcBf8C
[17] K. Kipper, A. Korhonen, N. Ryant, and M. Palmer, “Extending VerbNet with Novel Verb Classes”.
[18] M. Green, O. Hargraves, C. Bonial, J. Chen, L. Clark, and M. Palmer, “VerbNet/OntoNotes-Based Sense Annotation,” in *Handbook of Linguistic Annotation*, N. Ide and J. Pustejovsky, Eds., Dordrecht: Springer Netherlands, 2017, pp. 719–735. doi: 10.1007/978-94-024-0881-2_26.
[19] M. Marcus *et al.*, “The Penn Treebank: Annotating Predicate Argument Structure,” in *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. Accessed: May 18, 2023. [Online]. Available: https://aclanthology.org/H94-1020
[20] M. Palmer, D. Gildea, and P. Kingsbury, “The Proposition Bank: An Annotated Corpus of Semantic Roles,” *Computational Linguistics*, vol. 31, no. 1, Art. no. 1, Mar. 2005, doi: 10.1162/0891201053630264.
[21] “About FrameNet | fndrupal.” https://framenet.icsi.berkeley.edu/fndrupal/about (accessed May 18, 2023).
[22] V. Petukhova and H. Bunt, “LIRICS semantic role annotation: design and evaluation of a set of data categories,” p. 7.
[23] Q. Pradet, “Annotation en rôles sémantiques du français en domaine spécifique,” Thèse de Doctorat, Université Paris Diderot (Paris 7), France, 2015. [Online]. Available: https://hal.inria.fr/tel-01182711
[24] D. Gildea and D. Jurafsky, “Automatic Labeling of Semantic Roles,” *Computational Linguistics*, vol. 28, no. 3, Art. no. 3, Sep. 2002, doi: 10.1162/089120102760275983.
[25] M. Surdeanu, R. Johansson, A. Meyers, L. Màrquez, and J. Nivre, “The CoNLL 2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies,” in *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, Manchester, England: Coling 2008 Organizing Committee, Aug. 2008, pp. 159–177. Accessed: May 21, 2023. [Online]. Available: https://aclanthology.org/W08-2121
[26] K. M. Hermann, D. Das, J. Weston, and K. Ganchev, “Semantic Frame Identification with Distributed Word Representations,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore, Maryland: Association for Computational Linguistics, 2014, pp. 1448–1458. doi: 10.3115/v1/P14-1136.
[27] X. Lluís, X. Carreras, and L. Màrquez, “A Shortest-path Method for Arc-factored Semantic Role Labeling,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 430–435. doi: 10.3115/v1/D14-1049.
[28] R. S. Swier and S. Stevenson, “Unsupervised semantic role labelling,” in *Proceedings of EMNLP 2004*, D. Lin and D. Wu, Eds., Barcelona, Espagne: Association for Computational Linguistics, Juillet 2004, pp. 95–102.
[29] Q. Pradet, “Revisiting knowledge-based Semantic Role Labeling”.
[30] L. van der Plas and M. Apidianaki, “Cross-lingual Word Sense Disambiguation for Predicate Labelling of French,” in *Proceedings of TALN 2014 (Volume 1: Long Papers)*, Marseille, France: Association pour le Traitement Automatique des Langues, Jul. 2014, pp. 46–55. Accessed: May 21, 2023. [Online]. Available: https://aclanthology.org/F14-1005
[31] P. Exner, M. Klang, and P. Nugues, “Using Distant Supervision to Build a Proposition Bank”.
[32] J. Lang and M. Lapata, “Unsupervised Semantic Role Induction via Split-Merge Clustering,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 1117–1126. Accessed: May 21, 2023. [Online]. Available: https://aclanthology.org/P11-1112

[33] N. Garg and J. Henderson, “Unsupervised Semantic Role Induction with Global Role Ordering,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 145–149. Accessed: May 21, 2023. [Online]. Available: https://aclanthology.org/P12-2029

[34] I. Titov and A. Klementiev, “A Bayesian Approach to Unsupervised Semantic Role Induction,” in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Avignon, France: Association for Computational Linguistics, Apr. 2012, pp. 12–22. Accessed: May 21, 2023. [Online]. Available: https://aclanthology.org/E12-1003

[35] J. Materna, “Parameter Estimation for LDA-Frames,” in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia: Association for Computational Linguistics, Jun. 2013, pp. 482–486. Accessed: May 21, 2023. [Online]. Available: https://aclanthology.org/N13-1051

[36] D. Liu and D. Gildea, “Semantic Role Features for Machine Translation,” in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 716–724. Accessed: May 18, 2023. [Online]. Available: https://aclanthology.org/C10-1081

[37] P. Fung, W. Zhaojun, Y. Yongsheng, and D. Kai, “Automatic Learning of Chinese English Semantic Structure Mapping,” Jan. 2007, pp. 230–233. doi: 10.1109/SLT.2006.326797.

[38] D. Wu and P. Fung, “Semantic Roles for SMT: A Hybrid Two-Pass Model,” in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, Boulder, Colorado: Association for Computational Linguistics, Jun. 2009, pp. 13–16. Accessed: May 18, 2023. [Online]. Available: https://aclanthology.org/N09-2004

[39] A. Lakhfif and M. T. Laskri, “A frame-based approach for capturing semantics from Arabic text for text-to-sign language MT,” *International Journal of Speech Technology*, vol. 19, no. 2, Art. no. 2, 2015, doi: 10.1007/s10772-015-9290-8.

[40] M. Bazrafshan and D. Gildea, “Semantic Roles for String to Tree Machine Translation,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Sofia, Bulgaria: Association for Computational Linguistics, 2013, pp. 419–423. [Online]. Available: http://www.aclweb.org/anthology/P13-2074

[41] D. Trandabă\textcommabelowt, “Using semantic roles to improve summaries,” in *Proceedings of the 13th European Workshop on Natural Language Generation*, Nancy, France: Association for Computational Linguistics, Sep. 2011, pp. 164–169. Accessed: May 18, 2023. [Online]. Available: https://aclanthology.org/W11-2822

[42] “Effects of knowledge of semantic roles on summarizing written prose,” *Contemporary Educational Psychology*, vol. 6, no. 1, pp. 59–65, Jan. 1981, doi: 10.1016/0361-476X(81)90030-8.

[43] L. Suanmali, N. Salim, and M. S. Binwahlan, “SRL-GSM: A Hybrid Approach based on Semantic Role Labeling and General Statistic Method for Text Summarization,” *Journal of Applied Sciences*, vol. 10, no. 3, Art. no. 3, Mar. 2010, doi: 10.3923/jas.2010.166.173.

[44] S. Stenchikova, D. Hakkani-Tur, and G. Tur, “QASR: Question Answering Using Semantic Roles for Speech Interface,” 2006.

[45] D. Shen and M. Lapata, “Using Semantic Roles to Improve Question Answering,” in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, République tchèque: Association for Computational Linguistics, Juin 2007, pp. 12–21. [Online]. Available: http://www.aclweb.org/anthology/D/D07/D07-1002

[46] S. Narayanan and S. Harabagiu, “Question answering based on semantic structures,” in *Proceedings of the 20th International Conference on Computational Linguistics*, Genève, Suisse, 2004. [Online]. Available: http://www.aclweb.org/anthology/C04-1100

[47] M. Palmer *et al.*, “A Pilot Arabic Propbank,” in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08)*, Marrakech, Morocco: European Language Resources Association (ELRA), May 2008. Accessed: May 18, 2023. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2008/pdf/880_paper.pdf

[48] J. Mousser, “Classifying Arabic Verbs Using Sibling Classes,” in *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*, 2011. Accessed: May 18, 2023. [Online]. Available: https://aclanthology.org/W11-0142

[49] S. Elkateb *et al.*, “Building a WordNet for Arabic,” in *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC’06)*, Genoa, Italy: European Language Resources Association (ELRA), May 2006. Accessed: May 18, 2023. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2006/pdf/805_pdf.pdf

[50] “On the evaluation and improvement of Arabic WordNet coverage and usability | SpringerLink.” https://link.springer.com/article/10.1007/s10579-013-9237-0 (accessed May 18, 2023).

[51] “FrameNets In Other Languages | fndrupal.” https://framenet.icsi.berkeley.edu/fndrupal/framenets_in_other_languages (accessed May 18, 2023).

[52] N. Ghneim, E. Karhely, and W. Sa, “First Step of Building an Arabic FrameNet (AFN),” presented at the 13th International Business Information Management Conference, Marrakech, Maroc, 2009. [Online]. Available: https://www.researchgate.net/publication/284324899_First_Step_of_Building_an_Arabic_FrameNet_AFN

[53] M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki, “The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus,” in *NEMLAR Conference on Arabic Language Resources and Tools*, Cairo, Egypt, 2004.

[54] M. Diab, A. Moschitti, and D. Pighin, “Semantic Role Labeling Systems for Arabic using Kernel Methods,” in *Proceedings of ACL-08: HLT*, Columbus, Ohio, États-Unis: Association for Computational Linguistics, 2008, pp. 798–806. [Online]. Available: http://www.aclweb.org/anthology/P08-1091

[55] A. Kahina, “Un Système d’Annotation des Rôles Sémantiques dans la Langue Arabe Basé sur une Méthode d’Intelligence Artificielle,” Thesis, university center of abdalhafid boussouf - MILA, 2022. Accessed: May 18, 2023. [Online]. Available: http://dspace.centre-univ-mila.dz/jspui/handle/123456789/1899

[56] “Deep Learning Tutorials.” Laboratoire d’Informatique des Systèmes Adaptatifs, May 12, 2023. Accessed: May 18, 2023. [Online]. Available: https://github.com/lisa-lab/DeepLearningTutorials

[57] L. Deng and D. Yu, “Deep Learning: Methods and Applications,” *SIG*, vol. 7, no. 3–4, pp. 197–387, Jun. 2014, doi: 10.1561/2000000039.

[58] C. M. Bishop, “Neural networks and their applications,” *Review of Scientific Instruments*, vol. 65, no. 6, pp. 1803–1832, Jun. 1994, doi: 10.1063/1.1144830.

[59] “Artificial Neural Network Architectures and Training Processes | SpringerLink.” https://link.springer.com/chapter/10.1007/978-3-319-43162-8_2 (accessed May 18, 2023).

[60] K. Team, “Keras documentation: Layer activation functions.” https://keras.io/api/layers/activations/ (accessed Jun. 07, 2023).

[61] I. El Naqa and M. J. Murphy, “What Is Machine Learning?,” in *Machine Learning in Radiation Oncology: Theory and Applications*, I. El Naqa, R. Li, and M. J. Murphy, Eds., Cham: Springer International Publishing, 2015, pp. 3–11. doi: 10.1007/978-3-319-18305-3_1.

[62] "Deep Learning vs. Machine Learning: Beginner's Guide," *Coursera*, May 17, 2023. https://www.coursera.org/articles/ai-vs-deep-learning-vs-machine-learning-beginners-guide (accessed Jun. 12, 2023).

[63] "Inflectional Review of Deep Learning on Natural Language Processing | IEEE Conference Publication | IEEE Xplore." https://ieeexplore.ieee.org/document/8538416/similar#similar (accessed May 18, 2023).

[64] L. Màrquez, X. Carreras, K. C. Litkowski, and S. Stevenson, "Semantic Role Labeling: An Introduction to the Special Issue," *Computational Linguistics*, vol. 34, no. 2, pp. 145–159, Jun. 2008, doi: 10.1162/coli.2008.34.2.145.

[65] L. He, K. Lee, M. Lewis, and L. Zettlemoyer, "Deep Semantic Role Labeling: What Works and What's Next," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 473–483. doi: 10.18653/v1/P17-1044.

[66] "What is Python? Executive Summary," *Python.org*. https://www.python.org/doc/essays/blurb/ (accessed May 20, 2023).

[67] "Python (in Machine Learning)," *Corporate Finance Institute*. https://corporatefinanceinstitute.com/resources/data-science/python-in-machine-learning/ (accessed May 20, 2023).

[68] "Why Python is Good for Machine Learning," *Engineering Education (EngEd) Program | Section*. https://www.section.io/engineering-education/why-python-is-good-for-machine-learning/ (accessed May 20, 2023).

[69] "TIOBE Index," *TIOBE*. https://www.tiobe.com/tiobe-index/ (accessed May 20, 2023).

[70] "Quick start guide | PyCharm Documentation." https://www.jetbrains.com/help/pycharm/quick-start-guide.html#meet (accessed May 20, 2023).

[71] "Keras: Deep Learning for humans." https://keras.io/ (accessed May 20, 2023).

[72] "What is Tensorflow? Deep Learning Libraries and Program Elements Explained," *Simplilearn.com*. https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-tensorflow (accessed May 20, 2023).

[73] "pandas - Python Data Analysis Library." https://pandas.pydata.org/ (accessed May 20, 2023).

[74] "scikit-learn · PyPI." https://pypi.org/project/scikit-learn/ (accessed Jun. 12, 2023).

[75] "What is Tkinter used for and how to install this Python Framework?," *ActiveState*. https://www.activestate.com/resources/quick-reads/what-is-tkinter-used-for-and-how-to-install-it/ (accessed Jun. 12, 2023).

[76] Weischedel, Ralph *et al.*, "OntoNotes Release 5.0." Linguistic Data Consortium, p. 2806280 KB, Oct. 16, 2013. doi: 10.35111/XMHB-2B84.

[77] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang, "CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes," in *Joint Conference on EMNLP and CoNLL - Shared Task*, Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 1–40. Accessed: May 20, 2023. [Online]. Available: https://aclanthology.org/W12-4501