UNIVERSITÀ DI PISA

Dipartimento di Filologia, Letteratura e Linguistica

Corso di Laurea Magistrale

in Informatica Umanistica

Natural Language Inference and Transformers: a focus on predicate-argument structure

Relatore:
Prof. Alessandro Lenci

Candidato:
Marica Massari

ANNO ACCADEMICO 2020/2021

Contents

List of Figures

List of Tables

1. Introduction

In the recent years we assisted to a great progress of technologies dealing with natural languages. In particular, great results were achieved by machine learning models that were able to understand languages with an accuracy almost similar to the human one. The success of such models is due to the availability of computational resources as well as of large and high-quality datasets which enable them to understand the linguistic phenomena of the specific variety of language they are representative of. A recent switching point in the field of the Natural Language Processing was the development and the release of some pre-trained models and Transformers that significantly improves the state-of-the-art results for a number of different tasks.

Here, we decided to focus on the field of Natural Language Inference, a sub-task of the Natural Language Understanding that consists of analyzing the inference relation between a pair of sentences. The main goal of Natural Language Inference tasks is to say whether the first sentence (called premise) entails, contradict or is not related at all with the second sentence (the hypothesis). An increasing attention has been paid recently on this kind of tasks and consequently, some models, mainly of which are based on a Transformer architecture, allow to significantly improve the state-of-the-art results. However, these models often produce sentence representations that can not be easily understood by humans. Moreover, it is difficult to say which kind of information is actually included in those representations and in which specific part of them. For this reason, probing tasks are extremely useful in order to understand what information are really grasped by such models and unable us to make other progresses in Natural Language Processing.

With the development of this thesis we tried to put another brick towards this direction. We build an English dataset highlighting the inference relation between pairs of sentences, focusing on the predicate argument structure. Such dataset was used for some experiments. The main goal of these experiments was to analyze how some of the most famous freely available Transformer models are able to grasp about the inference relation between the premises and the hypotheses of our handmade dataset.

In chapter 2 of this thesis we give a view of the state of the art of Natural Language Processing and, more in details, of the field of Natural Language Inference. Then, in chapter 3 we describe what we mean as a predicate argument structure; while in chapter 4 we explain in details how the dataset was built. Finally, in chapter 5 we show the experiments setup, the models chosen and the results obtained over our dataset.

2. State of the art

In the last decade we assisted to an amazing development of new extremely powerful machine learning technologies that can handle many day by day activities of humans, involving different fields of application. Herein we are going to offer a general and brief view of some of these technologies handling human languages, called Natural Language Processing technologies. The idea of Natural Language Processing had emerged from the need for Machine Translation in the early 1940s, but it got a new life only when the idea of Artificial Intelligence emerged in 1960s. In the first stages of development, the main approach was to address linguistic problems with grammar based heuristics, so relying purely on symbolic, hand-crafted rules and underlining a deterministic approach to the language. Chomsky also identified some restrictive *regular grammars*, that were the basis of the regular expressions used to specify text-search patterns. However, such handwritten rules are very expensive and difficult to be drawn and they require a lot of effort by experts. Moreover, they are not sufficient to handle the complexity, ambiguity and unrestrictive nature of human languages.

Given these limitations of such approaches, in the 1980s the research had been guided towards the development of new data driven algorithms that do not need a strong theoretical linguistic framework as their starting point. This reorientation resulted in the birth of statistical Natural Language Processing, which was possible thanks to the availability of machine learning methods as well as the proliferation of large, annotated corpora employed to train and test machine learning algorithms. So, a new approach to human languages was born, also thanks to the application's spread of theories of the distributional structure of the language [47], along with the compositional theories of words meanings. Distributional theories of the language are based on the assumption that the statistical distribution of linguistic items in context plays a key role in characterizing their semantic behavior [1]. Distributional models build representations of the words' meaning by extracting co-occurrences from corpora. Starting from this, it was possible to create *Word Space Models* (VSM), which are computational models of word meaning based on the way words occur

in different contexts. Word Space Models, that come from the information retrieval [43], represent words with a n-dimensional vector, called *word embedding*, that can be visualized in a n-dimensional space. This kind of representation resulted to be very successful because it ensures that words that are similar appear to be near in the space. Specifically, these geometrical representations allow to compute a similarity measure between words.

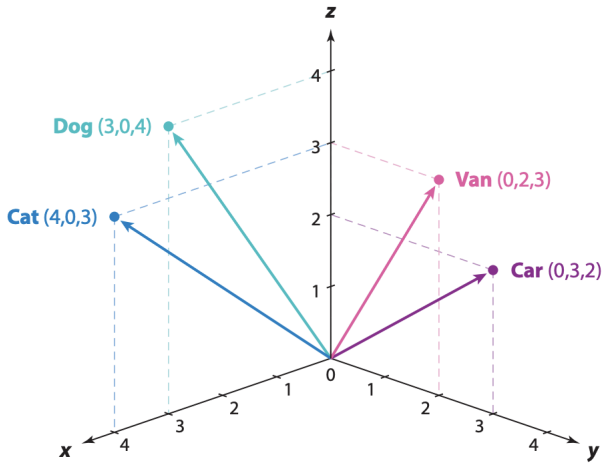

Figure 2.1: Distributional vectors of some lexemes, visualized in a 3-dimensional space.

Therefore, distributional representations are geometrical representations, graded and distributed because information is encoded in the continuous values of vector dimensions, in contrast to the discrete and categorical representations of words that comes from symbolic semantic approaches. In the following sub-section we will assess a more detailed view of the way word embeddings are produced and used by some of the most known models.

2.1 Word embeddings

The way data is encoded is a crucial aspect for data analysis in general. Categorical data are input features that represent one or more discrete items from a finite set of choices. Such type of data is efficiently represented via sparse vectors, which are vectors with very few non-zero components, as word embeddings are. Word embeddings rely on the distributional hypothesis according to which co-occurring words are semantically related. A basic example of the creation of word embeddings is to consider a word in all its contexts

(where a context is the set of words that appear within a fixed-size window) and count how many times it co-occurs with all of them. We can easily create a list of numbers, each of which corresponding to the times we have seen the target word *t* with a certain context word *c*: this list of numbers can be represented as a vector. Of course, this kind of vectors are as long as the size of the vocabulary (*V*) of the corpus used for generating them. This is the reason way these vectors are sparse. Then, we can compute a *word-context matrix*, a matrix of dimension V*V that counts the frequencies of co-occurrence of words in a collection of contexts [34]. An example of co-occurrence matrix is shown in table 2.1.

	Cat	Pig	Van	Car
Cat	0	0	2	3
Pig	0	0	4	1
Van	2	4	0	5
Car	3	1	5	0

Table 2.1: Example of co-occurrence matrix.

In order to avoid gargantuan sparse words representation that are not particularly efficient within a machine learning system, researchers found many ways to obtain a dense representation by extracting relevant word features from the sparse vector, in such a way that the noise is reduced. There are different ways of doing so: global matrix factorization methods such as and the *Singular Value Decomposition*; or Neural Network models like *Word2vec* and *Glove* [32]. In this work we give an overview of these last models, focusing on how they work and what their main pros and cons are.

In 2003, Bengio and some of his colleagues presented a way to learn a distributed representation for words which allows to fight the so-called problem of the *curse of dimensionality* [4]. The curse of dimensionality is a fundamental problem that makes language modeling difficult to face. This is based on the assumption that a word sequence on which the model will be tested is likely to be different from all the word sequences that have been seen during the training phase. For example, if one wants to model the joint distribution of 10 consecutive words (so 10 discrete random variables) in a natural language with a vo-

cabulary of size V equals to 100.000, there are potentially $100.000^{10} - 1$ free parameters. On the contrary, when we model continuous variables it is easier to obtain a good generalization because the function to be learned can be expected to have some local smoothness properties. So, they proposed a new approach to face the curse of dimensionality using distributed representations.

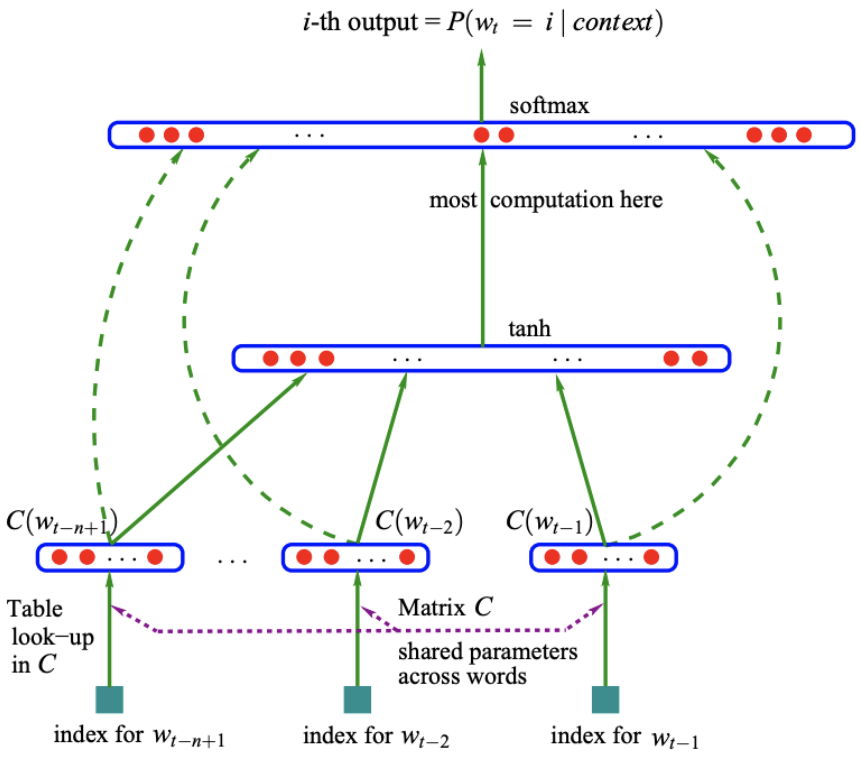

Figure 2.2: Neural network architecture of the langugage model proposed by Bengio et al. in 2003.

Each word in the vocabulary is associated with a *word feature vector*, in which the number of features is much smaller than the size of the vocabulary. Similar words are expected to have a similar feature vectors. The joint probability function of word sequences is expressed in terms of the feature vectors of these words in the sequence. This function has parameters that can be iteratively tuned in order to maximize the log-likelihood of the training data, eventually adding also a weight decay penalty as a regularitazion criterion.

Because the probability function is a smooth function of the feature values, a small change in the features of a word will induce a small change in the probability. The feature vectors can be learned or initialized using some prior knowledge.
They performed two experiments on different corpora and they showed that the proposed approach yields much better perplexity than the current state-of-the-art methods. In some of the modern models, the same foundation posed by Bengio et al. is still used. One of the most popular model that exploits this kind of approach is Word2vec.

Word2vec is a framework for learning continuous word vectors from huge datasets [2]. It introduces some simplification, both on the model and on the way of computing the representations, in order to guarantee a faster training but also a higher accuracy. There exists two versions of this framework: CBOW (Continuous Bag of Words) and SkipGram. CBOW is trained on a large corpus to predict a target word given its context. As the name of the model suggests, the context of a target word is given by the words that stands within the window of size m. CBOW produces a binary solution for each single word. SkipGram is the opposite of CBOW because it is trained to predict context words within a window of size m, surrounding a target word t. The advantage of these frameworks is that the networks can be trained on any text without the need of human-labeled data.

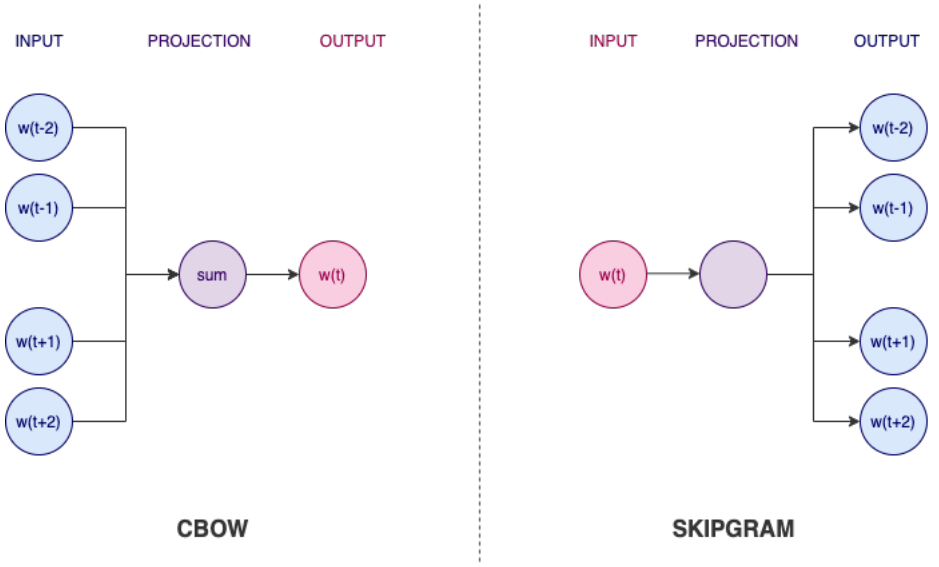

Figure 2.3: Flow of information on CBOW (on the left) and SkipGram (on the right).

In 2014, Glove (GLObal VEctors for words representation) was released [40]. The model they proposed relies on global statistics of word occurrences in a corpus to produce a vector space with meaningful linear substructure. The basic idea of Glove is that the appropriate starting point for word vector learning should be with ratios of co-occurrence probabilities rather than the probabilities themselves. This ensures Glove to be a global log-bilinear regression model for the unsupervised learning of word representations that outperforms other models on word analogy, word similarity, and named entity recognition tasks.

These models demonstrated to be able to encode both syntactic and semantic features of words. However, the main problem is generating the embedding of a word that takes into account the specific context of the word to define its meaning. The real shift in the way word embeddings are generated was in the 2018, thanks to the development of Transformers and pre-trained models. In fact, while previous models produce a word representation that is static, researcher started to focus on models that were able to generate contextual representations, which means that the same word has a different word embedding if it occurs in different contexts.

2.2 Sentence embeddings

Another key problem in all Natural Language Processing tasks is how to represent a whole sentence in a way that it can be processed by a machine learning model. In this sub-section, we are going to give a quick overview of how to produce sentence embeddings, without aiming to give a complete explanation.
The easiest method to address this problem is to sum the vectors representing all the words of the sentence, in such a way that a single vector is produced and it somehow entails the information deriving from the words that compose the sentence. We refer to this approach as *Bag-Of-Words approach* (BOW) [39].
A more sophisticated approach were built using unsupervised models like Skip-Thoughts [11]. Skip-Thoughts is an encoder-decoder model based on a recurrent neural network

(similarly to SkipGram) which produces generic sentence representations. Such sentence representations, called Skip-Thoughts Vectors, have been proved to be robust and perform well in practice.

The next turning point on the development of approaches producing sentence representations was the release of *Supervised Learning of Universal Sentence Representations from Natural Language Inference Data* by Conneau et al. [6]. They demonstrated that using the SNLI corpus (The Stanford Natural Language Inference) [5] as training set (with a BiLSTM and with the technique of Max pooling) it is possible for a network to better learn universal sentence representation. This approach consistently outperform unsupervised methods like Skip-Thought vectors on a wide range of transfer tasks. They explored various architectures: standard recurrent models (such as LSTMs and GRUs) with both mean and max-pooling over the hidden representations; a self-attentive network that incorporates different views of the sentence; and a hierarchical convolutional network. They found that the architecture producing the best current universal sentence encoding methods was a BiLSTM network with max pooling.

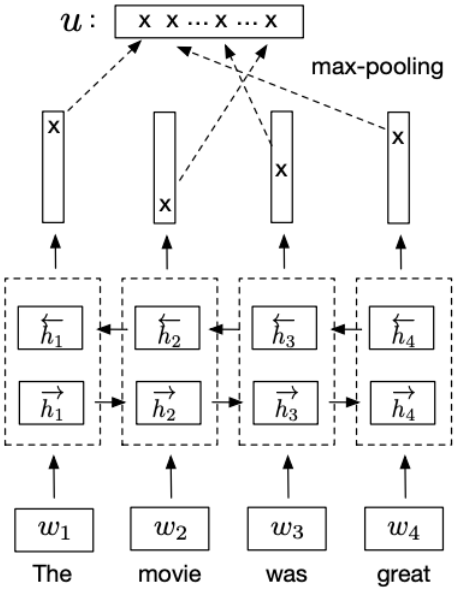

Figure 2.4: Bi-LSTM max-pooling network

2.3 Transformers

Transduction problems and sequence modeling tasks have usually been addressed by the NLP community using recurrent neural networks, long short-term memory and gated recurrent neural networks until the establishment of a new type of architecture that is the Transformer. The Transformer is a simple network architecture based soleley on the attention mechanism, dispensing with recurrence and convolutions entirely [20]. The attention is a mechanism that allows the model to selectively concentrating on a few relevant things, while ignoring others.

The Transformer was introduced in 2017 with the well-known paper *Attention is all you need*. It has an encoder-decoder structure using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder. The process is auto-regressive at each step because the model consumes the previously generated symbols as additional inputs when generating the next one. The input of the model is a text, where each word is represented as a word embedding. Its summarized architecture is shown in figure 2.5.

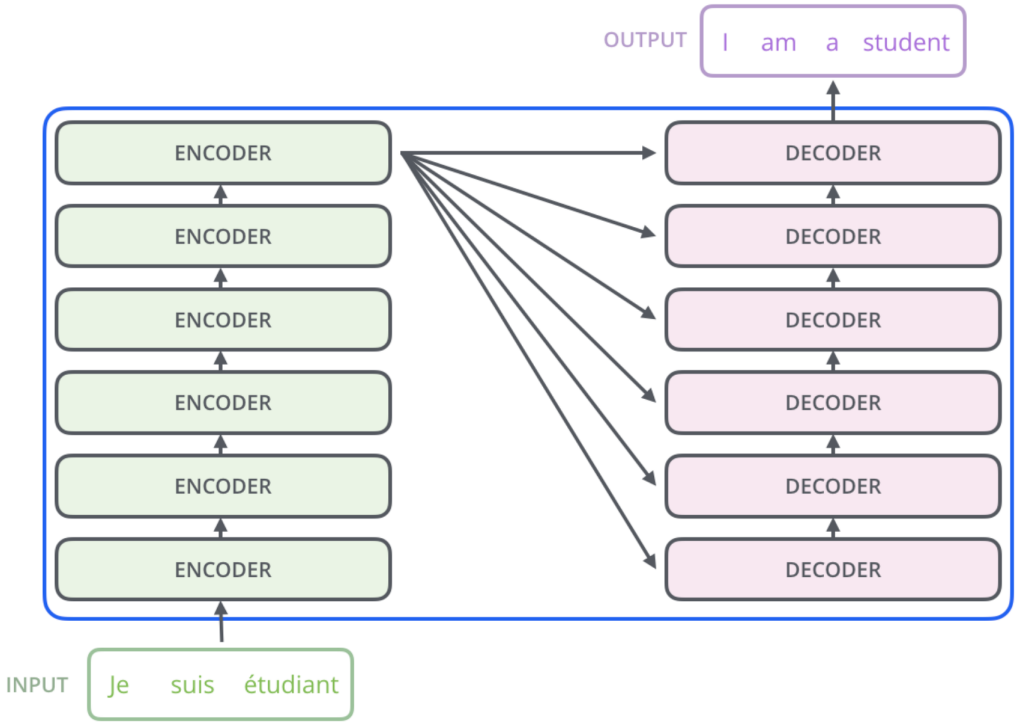

Figure 2.5: Architecture of the Transformer.

The encoder is composed of a stack of $N = 6$ identical layers, each of which has two sub-layers. The first sub-layer is a *multi-head self-attention mechanism*, while the second one is a simple, position-wise fully connected feed-forward network. A residual connection is employed around each of the two sub-layers, and it is followed by an *add & normalization* layer. All sub-layers in the model, as well as the embedding layers, produce outputs of dimension d equals to 512.

The decoder is composed of 6 layers too but, in addition to the two sub-layers, it also has a *masked multi-head attention* sub-layer that performs multi-head attention over the output of the encoder stack. Also in the decoder a residual connection is employed around the sub-layers, followed by the normalization layer. The masked multi-head attention sub-layer ensures that the predictions for position i can depend only on the known outputs at earlier positions. The output of the decoder is then processed by a linear layer which is a fully connected neural network capable of producing a *logits vector* filled with the probabilities for every word in the dictionary. The most probable word is then selected by the *softmax layer*.

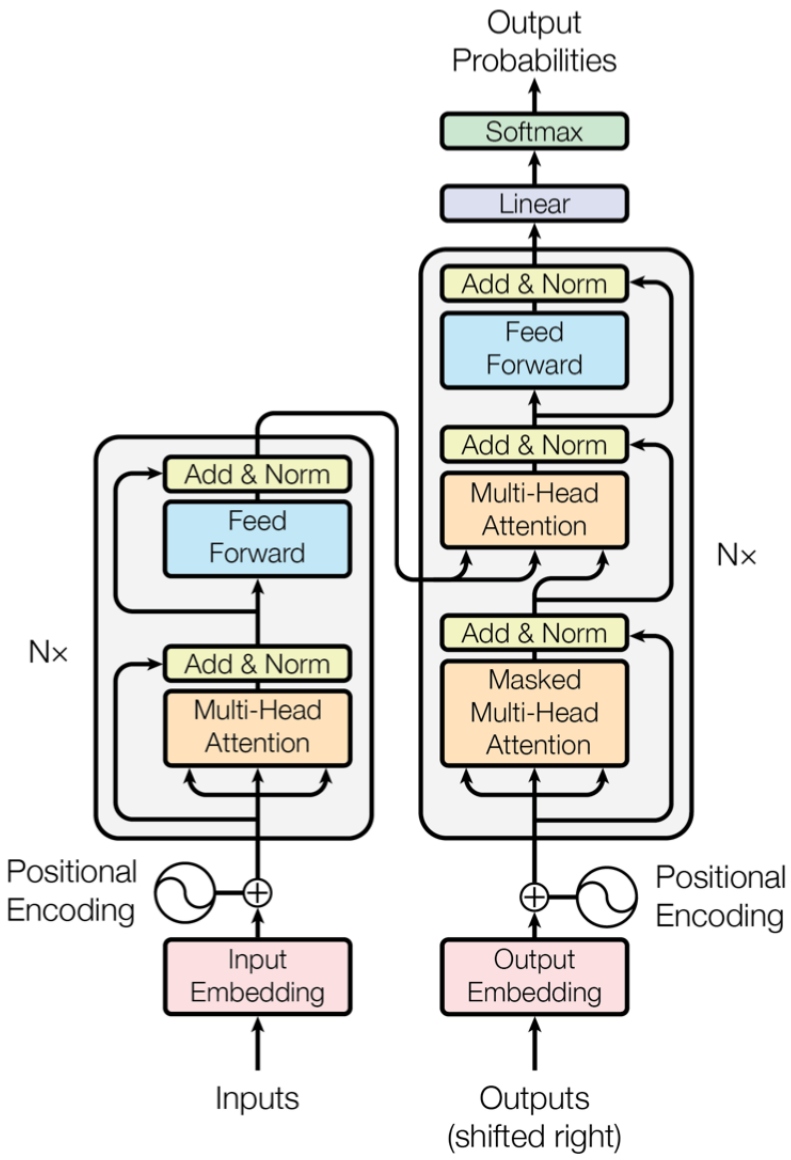

Figure 2.6: Components of the Transformer.

As suggested by the title of the introductory paper, one of the key element of Transformer's success is the central role of the attention mechanism that allows the machine to better store the information regarding the words, inserting information about the surrounding ones into the encoding of the single word taken under consideration. The attention mechanism consists of a mapping function between some numerical vectors: a query (Q) and a set of key-value pairs (respectively K and V) are mapped into a single output. So for each word in the input, three vectors of dimension 64 are created by multiplying

the input embedding (of dimension 512) by three matrices resulting from the training process. The output of the attention function is a weighted sum (that is an additive summary) of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. So, the result we obtain is a fixed-size representation of an arbitrary set of representation (the values), dependent on some other representation (the query).

There exist several variants of the attention mechanism. In the multi-head attention several attention layers run in parallel and then the output of each of them are concatenated together and projected as shown in figure 2.7. The multi-head attention allows the model to jointly attend to information from different representation sub-spaces at different positions while, with a single attention head, averaging inhibits this.

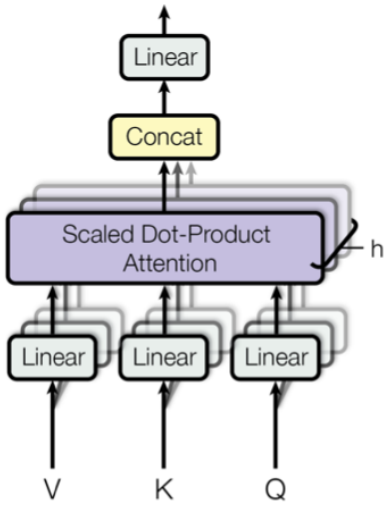

Figure 2.7: Multi-head attention mechanism.

Vaswani et al. showed that, for translation tasks, the Transformer can be trained significantly faster than architectures based on recurrent or convolutional layers. Self-attention layers are faster than recurrent layers when the sequence length is smaller than the representation dimensionality. Thanks to the freely availability of the code they used to train and evaluate their models, many researchers put a lot of efforts on the development of such architectures. The effectiveness of transformers has also been proved by using them in the process of pre-training, which will be explained in details in the next sub-section of this thesis.

2.4 Pre-trained models

The development of deep learning has led to great successes into different fields of application. Deep learning models automatically learn low-dimensional continuous vectors from data as task-specific features. However, even if these models are usually able to achieve a high accuracy on the test set, they will fail if the dataset they are trained on is not sufficiently large. Indeed, deep neural networks usually have a large number of parameters that make them overfit on small training dataset, not being able to achieve good generalization capabilities. Moreover, they require a lot of time and huge computational power in order to be trained. Considering these issues, massive efforts have been devoted to manually construct high-quality datasets. However, it is extremely time-consuming to manually annotate large-scale data.
So recently, researchers focused on the development of pre-trained models, exploiting the so-called transfer learning mechanism that has been shown to be effective for improving many Natural Language Processing tasks [22]. The main idea is that previously learned knowledge can be used to solve a new task that has some features in common to the ones already faced. Pre-trained models follow two development steps:

- Semi-supervised training on a large amount of data, that enables the model to grasp patterns in the language. At the end of this process, the model can obtain some generic word embeddings from the unlabeled data seen.
- Supervised training on a specific task using a labeled dataset.

Language representations learned by such models at the end of the pre-training phase can be applied to downstream tasks by using two different strategies: *feature-based* and *fine-tuning*.
The feature-based approach (typical of ELMo, Embdeddings from Language Models) uses task-specific architectures that include the pre-trained representations as additional features [15]. In the specific case of ELMo, the bidirectional long short term memory

network on which it is built, ensures to store a lot of information deriving from the context of the word. The word embedding generated by ELMo are contextual embeddings that contains also morphological features.
On the other hand, the fine-tuning approach (such as the one of OpenAI GPT) introduces minimal task-specific parameters, and then the model is trained on the downstream tasks by simply fine-tuning all pre-trained parameters [16]. Both the two approaches use unidirectional language models during the pre-training to learn general language representations. It is the main drawback of language model pre-trained decoders such as ELMo and GPT because natural language is a complex system which requires that each token has to be processed and analyzed by taking into account context from both the two directions in order to fully understand the whole text.

2.4.1 BERT

A switching point on the development of such pre-trained models was the release of BERT by Google [8]. BERT stands for Bidirectional Encoder Representation from Transformers and, unlike previous models, it considers bidirectional context in all encoder layers (also called *transformer blocks*). So, the architecture of BERT is a multi-layer bidirectional Transformer encoder based on the original implementation described in Vaswani et al. (2017) and released in the tensor2tensor library [1]. There are basically two versions of this architecture, varying on the size, number of attention heads and parameters:

- *BERT base* with 12 transformer blocks, 12 self-attention heads and a hidden size equals to 768 (for a total number of parameters = 110M). It has the same model size of OpenAI GPT so comparisons between the two are possible.
- *BERT large* with 24 transformer blocks, 16 self-attention heads and a hidden size equals to 1024 (for a total number of parameters = 340M).

A distinctive feature of BERT is its unified architecture across different tasks. Also there is just a minimal difference between the pre-trained architecture and the final downstream

[1] Freely available at the following link https://github.com/tensorflow/tensor2tensor

architecture.

BERT takes as input a sequence of words: the first token of every sequence is the special classification token [CLS]. The pre-trained objective that BERT uses is a masked language model, inspired by the Cloze task [36]. The masked language model randomly masks the 15% of the tokens from the input. Then the model is used to predict the original masked token, using the cross entropy loss as objective function [30]. The masking phase of the sentence is illustrated in the figure 2.8.

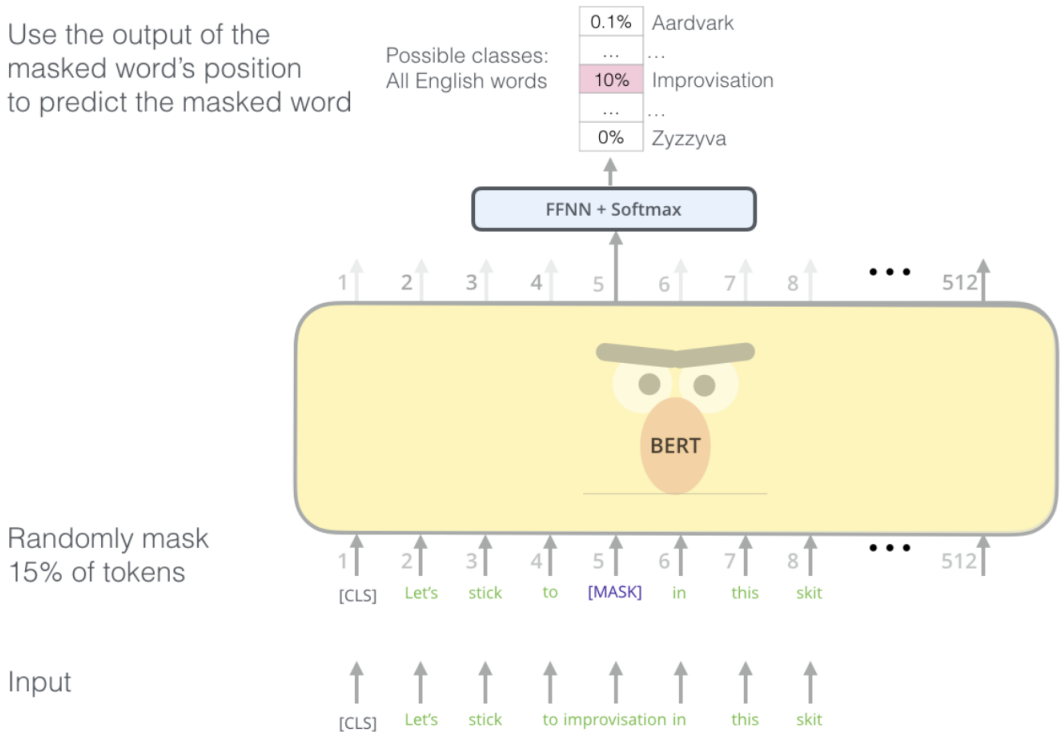

Figure 2.8: Example of the BERT masked language model.

In addition to this, another unsupervised task is also used for pre-training BERT: the ”next sentence prediction” binarized task jointly pre-trains text-pair representations. Specifically, when choosing the sentences A and B for each pre-training example, 50% of time B is the actual next sentence that follows A, and 50% of the time it is a random sentence from the original corpus. This latter task is essential to capture sentence relationships, and therefore it allows BERT to be effective also for tasks like Question Answering and Natural Language Inference.

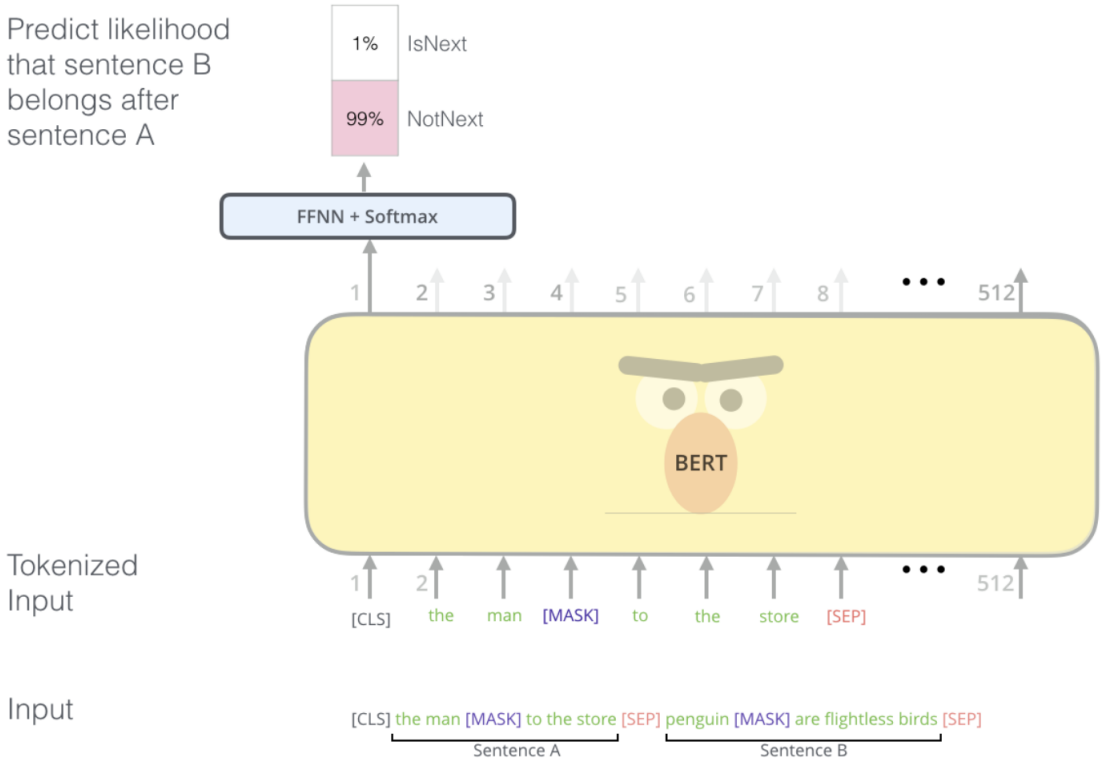

Figure 2.9: Example of the BERT's next sentence prediction task.

As a result of these two procedures, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of both sentence-level and token-level tasks, without substantial task-specific architecture modifications.

2.5 Probing tasks

A key point when facing whatever Natural Language Processing task is understanding what kinds of information are stored on words and sentences representations, and not simply just the accuracy obtained by the model that was used. Downstream tasks are sometimes used to evaluate the quality of the representations learned by the model. However, they require complex forms of inference and this makes it difficult to use them for analyzing the quality of the information encoded. This is why it has become common to create ad hoc tasks called *probing tasks*. A probing task is a classification problem that focuses on simple linguistic properties of sentences, minimizing interpretability problems.

Because of their simplicity, it is also quite easy to control biases. In practice, a model that was already trained for a specific task is used to produce the embeddings. Then, such embeddings are used to train a classifier that categorizes a specific linguistic phenomenon. If the accuracy obtained with the newly trained classifier is good, it means that the base model used is able to store into its representations the information required by the chosen task. Different linguistic properties can be probed, including semantic roles [28], negation scopes, constituents and part-of-speech tags [44], syntactic properties like the sentence's length and words' orders [3], agreement information [9] and the tense of the main clause [25]. Especially with the development of contextualized embeddings, have become popular to apply probing tasks to word-level contextual representations, attention mechanisms and other syntactic knowledge.

One of the first usage of a probing task was applied to the semantic information detected by models. Ettinger et al. (2016) constructed a sentence dataset to capture a wide variety of syntactic structures and configurations. Such dataset is used for addressing a classification task over two types of semantic information: semantic roles and negation scopes.

Adi et al. (2017) investigated the capacity of models to encode the sentence length, the items within it and the order of the words. They trained a classifier that was able to predict specific properties of the text analyzed given the sentence represented trough a vector. Their base assumption was that if the model is not able to predict the features, then the required information are not encoded into the input vectors. Given a sentence embedding, one goal was to predict the length of the sentence expressed in number of words. To do this, they trained a multi-class classifier with eight classes corresponding to range of length (e.g 5-8 is the class indicating a number of words between five and eight). Another point was about investigating if a word w is present in the sentence s. So, they trained a binary classifier that takes in input a sentence and a word and tells if the word is contained or not in the sentence. Finally, they trained another binary classifier in order to probe the word ordering in a sentence. In this case, the input of the classifier is a concatenation of three vectors: the classifier is called to predict whether the word a appears before the word b in the sentence s.

More recent applications of probing tasks are about investigating the linguistic knowledge

embedded by contextual word embeddings. For example, Liu et al. (2019) investigated the transferability of contextual word representations trough sixteen different probing tasks. Six of them examined how the models handle the task of token labeling (like recognize syntactic roles, disambiguate of Part-Of-Speech tags, determine lexical-semantic contribution of prepositions, label sentence with the factuality of the events described etc.). The remaining tasks probe different kind of relationships between words: for example they aim to distinguish between *arc prediction* (a binary classification task trained to identify if exists a relationship between two words) and *arc classification* (a multi-labeled classification task trained to understand which kind of relationship subsists between two tokens given as input). Both the two tasks were developed for syntactic and semantic relationships
Many other works of this kind are available such as the ones by Tenney et al. (2019) [19], Jiang and De Marneffe (2019) [31], Richardson et al. (2019) [18] etc. The purpose of this brief overview was to show how probing tasks are developed and their importance for examining whether particular language properties are sufficiently encoded into representations. This is a crucial aspect nowadays when using sophisticated models that produce state-of-the art results in many application fields but representations that are difficult to understand and to be examined by humans.

2.6 Natural Language Inference

A branch of the Natural Language Processing is the Natural Language Understanding. For the purpose of this work, we focused our attention on a specific task: the Natural Language Inference (NLI), which is one of the standard benchmark tasks for Natural Language Understanding [24]. NLI is the task of identifying whether a sentence can be inferred from, contradicted by, or not related to another sentence. The first sentence is usually called *premise* while the second one *hypothesis*. These tasks were originally named also *Recognizing Textual Entailment*. This name was introduced for the first time by Dagan and Glickman (2004) who illustrated the textual entailment phenomenon and a generic model for capturing it by using a ”shallow” level of semantics [26]. In its original formulation, the task consists of establishing whether the meaning of the hypothesis can be inferred

from the premise text. The set of sentences under examination were created ad hoc for the task, addressing in particular the common-sense and the common-knowledge of the language. In the past decades there has been a surge of interest in NLI due to the fact that it focuses on crucial aspects for a number of different application fields such as question answering, information retrieval, semantic search, natural language generation and automatic text summarization. In fact, the importance of this task can be understood by thinking that a system that is not able to identify the implications of a sentence is not able to capture the overall sense of the sentence itself [38]. This explains the proliferation of good quality corpora developed for addressing this kind of tasks (like the Stanford Natural Language Inference corpus and the Multi-genre Natural Language Inference corpus).
The problem of NLI has been formulated in a number of different ways by many researchers, even if it always requires to pairwise consider two texts: a premise P and a hypothesis H. The definition of datasets for NLI typically starts by asking some annotators to write some hypotheses sentences for each of the premises sentences already given. Each pair of sentences has to be labeled as *Entailment*, *Neutral* or *Contradiction*. However, there exist different kinds of datasets used for addressing tasks of NLI that vary on sizes, sources of texts, types of the considered linguistic phenomena, annotation strategies and so on. In the following sub-section we will see some of them.

2.6.1 Corpora and benchmark datasets

One of the first datasets of this type was created ad hoc for the first PASCAL Recognising Textual Entailment Challange (RTE-1) [27] [7]. The RTE-1 dataset consists of manually collected text fragment pairs. The participating systems were required to judge for each pair of texts whether the first entails the second one: so it was a binary classification task. The pairs were representative of inferences in various application types (that they termed *tasks*), such as Question Answering, Information Extraction, Information Retrieval and Machine Translation. This challenge raised noticeable attention in the research community, such that it was followed by many workshop and shared tasks with the same focus point (for example an ACL 2005 Workshop on Empirical Modeling of Semantic Equivalence and Entailment, some SemEval tasks across different years as well as other

interesting works). It generated a number of high-quality, hand-labeled datasets but that were usually small in the size.

The first large NLI corpus is the Stanford Natural Language Inference (SNLI) (Bowman et al.; 2015). [5] It was composed by 570k human-written English pairs of sentences, manually labeled as entailment, neutral or contradiction. The great increase in the size with respect to the previous available corpora allows a neural network based model to perform competitively on natural language inference benchmarks for the first time. The premises were collected form the Flickr30k corpus, a collection of captions of images, while the hypotheses were written by some users trying to keep the dataset balanced. However, the fact that the sentences in the SNLI are limited to descriptions of concrete visual scenes renders these pairs of sentences short and simple. This make irrelevant to the task performance to handle many key phenomena (like tense, belief and modality). Because of this, the SNLI corpus is not sufficiently demanding to serve as an effective benchmark for NLI.

Another interesting corpus is the Multi-Genre Natural Language Inference (MultiNLI) [46]. It was the first NLI corpus that makes it possible to develop and evaluate machine learning models for sentence understanding on nearly the full complexity of the language. Indeed, it contains 433k pairs of sentences that represent both written and spoken speech, along with a wide range of styles, degrees of formality and topics. The premise sentences of these pairs are derived from ten different sources of american english texts: nine of these sources come form the second release of the Open American National Corpus (OANC; Ide and Suderman; 2006); while for the last one they used eight freely available works of contemporary fiction of different genres, written between 1912 and 2010. Then, they presented a crowdworker with a premise sentence (with a minimal preprocessing) and it was asked them to compose three novel sentences (the hypotheses), forcing the data to be balanced among the three classes (entailment, neutral and contradiction).

Thanks to the increasing number of freely available large corpora , a lot of fine-tuned models trained for recognizing textual entailment are also available. Most of them obtain very good results on the test set derived from the datasets. Nevertheless, it was pointed out by many researchers that this high accuracy scores depend on the data on which the models were trained, while they showed poor performances whenever the kind of textual

data would come from another source [37]. This highlights the well-known problem of the domain adaptation, typical of many machine learning models due to their generalization capabilities.

Finally, another benchmark dataset for NLI (but also for other Natural Language Understanding tasks) is GLUE (Wang et al.; 2018). GLUE stands for General Language Understanding Evaluation and it consists of a benchmark for a collection of NLU tasks including question answering, sentiment analysis and textual entailment [21]. GLUE is centered on nine English sentence understanding tasks, which cover a wide range of domains, data quantities and difficulties. Four tasks of these are inference tasks based on the MultiNLI, QNLI (Rajpurkar et al.; 2016) [17], RTE and WNLI (The Winograd Schema Challenge). GLUE (but also also its successor SuperGLUE (Wang et al.; 2019)) includes an associated online platform for model evaluation, comparison and analysis. Moreover, in order to understand the types of knowledge learned by models and to encourage linguistic-meaningful solution strategies, GLUE also includes a set of hand-crafted analyses examples for probing trained models. While, the GLUE benchmark mostly reflects an application-driven distribution of examples, their diagnostic dataset include small and manually-curated test set. Each diagnostic example is an NLI pair of sentences with an associated tag for the linguistic phenomena demonstrated. This ensures to analyze which kind of errors the models commit, and therefore which sub-features of the language the models are able to grasp and which they do not. While building such dataset they tried to cover a variety of linguistic phenomena that are divided into four broad categories. Lexical semantics, Predicate-argument structures, Logic and Knowledge. Since these categories are vague, they divided each of them into some fine-grained categories.

The approach of GLUE was inspirational for the work described in this thesis. More specifically, we decided to focus on a quite big category of linguistic phenomena that is the predicate-argument structure. The predicate-argument structure is about how the parts of a sentence are composed together into a whole. A detailed view of this kind of phenomena is reserved on the next section of this thesis.

3. Predicate argument structure

The coarse-grained category of linguistic phenomena taken under consideration in this thesis is the predicate-argument structure. Predicate-argument structure is about how words are combined together into the whole sentence and around the main verb. So, since this vague definition of a predicate-argument structure, we decided to identify more specific categories regarding both syntactic and semantic aspects. For each of these fine-grained categories we identified some examples. First of all, we want to give a general overview of what a predicate-argument structure is, looking at how its definition and representation changed over years. Then, we are going onto more details, giving some examples.

3.1 Compositional theory

The definition of the predicate-argument structure has its basis on the compositional theory of the language (aka Fregean Compositionality) introduced for the first time by Frege in 1923. The development of the compositional theory represents a key point on the history of linguistics and more precisely of semantics: some of the modern semantic theories still refers to this principle, even if they apply some adjustments [29].
Compositionality is defined as an essential property of complex linguistic expressions. According to this theory, a complex expression is compositional in the sense that its meaning is determined by the meaning of all its component parts and their syntactic arrangement. All the elements of the meaning of a sentence can be found on the meaning of the lexemes composing the sentence itself. This principle is a key point for understanding how we can build and comprehend a potentially infinite number of combinations of a finite set of different words, just knowing the meaning of all of them and following some specific syntactic rules that are language-dependent. The main ingredients of the compositionality are a repertoire of stored linguistic units associated with a meaning (lexical items and semantic memory) along with the combinatorials principles used to build the interpretation of complex expressions (grammar and compositional semantics). The lexical items include

a wide range of linguistic expressions varying in size and degree of fixedness, but they all need to be associated to a meaning and memorized.

The formal implementation of the compositional principle is possible due to the definition of two types of parts composing a linguistic complex expression: the functional expression and its argument (or arguments). The arguments are complete in themselves, while the functions can be seen as the "unsaturated" part that need to be filled with at least an argument in order to be properly semantically processed. So a process of saturating the variable of a function is needed to process an expression. When an argument is applied to an interpretation function we refer to it as an assignment function (or *function application*). The application of an argument to a predicate does not change the meaning of the argument itself. When an argument for which we know the meaning is applied to its functional part, we are able to compute the value of the expression. For example, let's consider the phrase *Tom chases Jerry*. First of all, the meaning of this sentence can be inferred from the meaning of all its parts. Than, if we know all the elements, we can say that the predicate *chases* requires two arguments, so two function applications are needed in order to derive the interpretation of the sentence. Firstly, we apply the function *chases(arg1, arg2)* to the argument *Jerry* such that we get the function *chases(args1, Jerry)*. Then, we apply *chases(args1, Jerry)* to the argument *Tom*, in order to get *chases(Tom, Jerry)*.

The representation of the predicate-argument structure is a key concern in linguistic studies since the very early days of modern linguistics. In order to talk about language function application, it has been introduced the notion of *lambda*. Lambda λ is the name given to the operator that introduces placeholders for missing semantic components in an expression, which are represented as variables [42]. So, every λ-expression is a function looking for an argument to act on. In general, we can represent an application function with the following function

$$\lambda x[f(x)] : A \rightarrow B$$

that maps an argument a of type A to one and only one element b belonging to the type B, called the *value* for the function given the argument a. Let's suppose we have to analyze the expression *Jake sleeps*, we can formally express it as it follows:

$$\lambda x[sleeps(x)](Jake) \Rightarrow sleeps(Jake)$$

Another key aspect of the Fregean compositionality is about the semantic types of the arguments that are applied to a specific predicate. According to Frege, a predicate asks for some specific semantic types. So, in order to say that a complex expression is well-formed, the arguments applied to the function must have the appropriate semantic types. As stated by Pustejovsky and Batiukova (*Lexicon*; 2019), this requirement is actually built into the rule of function application stated as: *"A predicate β is an unsaturated expression, which, when combined with its argument, α, becomes a saturated expression, $\beta(\alpha)$. Formally, if the argument α is of type a, and the function β is of type $a \rightarrow b$ (i.e., if β maps expressions of type a into expressions of type b), then $\beta(\alpha)$ is of type b"*. The type a $\rightarrow$ b is denoted as *derived type* as it is built up from types *a* and *b* (that can be seen as a sort of input) with the application of the *type constructor*, represented by the arrow $\rightarrow$. The claim about the selection of specific semantic types of arguments is a controversial aspect of the compositional theory, because it is considered to be too strict by some other linguists.

3.2 Model-theoretic semantic theories

After the definition of the Fregean compositionality, a lot of formal theories of meaning adopted that main principle. More specifically, some theories have been developed based on the compositionality and the definition of a model. That model is a structured copy of the world that enables us to check whether a linguistic expression translated into a formal language is true or false. This type of theories are called *model-theoretic*. One of the most influential model-theoretic semantic theory is the Montague semantic introduced in the 1970s.

According to Montague, we can define the truth condition of a sentence if it is firstly translated into a formal language. Unlike natural languages which are all ambiguous, a formal language is unambiguous and the sentence can be easily represented trough the predicate argument structure. Such sentence can be interpreted by the model in one and only one way: the meaning of that sentence is given by its truth conditions [45]. In the Montague's view, when an expression is translated into a formal language a semantic type is assigned

to it. There can be three different kind of semantic types: individual entities e, truth values t, and functional types $\langle e, t \rangle$. A function f of type $\langle e, t \rangle$ is a function f: D $\rightarrow$ (0, 1), where D is a set of individuals and (0, 1) is the range of possible truth values (which are boolean values).

An essential feature of Montague semantics is the systematic relation between syntax and semantics: the typed function application is the semantic operation that works in parallel with the syntactic rules of the grammar. The syntactic rules derive the compositional translation of a sentence into the logical form that represents its truth conditions. Therefore, such rules are operations which act on inputs and yield an output. If an expression E is the output of the application of a rule R, then the inputs that form E are defined as being the parts of E in that derivation. All the parts that play a role in the syntactic composition of a combined expression, must also have a meaning. Semantic and syntax works in parallel in the sense that there are semantic rules that operate on input meanings and yield an output meaning. So, each syntactic rule must be accompanied by a corresponding semantic rule which says how the meaning of the compound is obtained. For example, let's consider the sentence *Tom chase Jerry*. The derivation of its logical form can be obtained by considering both syntactic and semantic aspects of the sentence, as shown in the figure 3.1. This tree does not depict the constituent structure of the sentence: it shows how the sentence is formed.

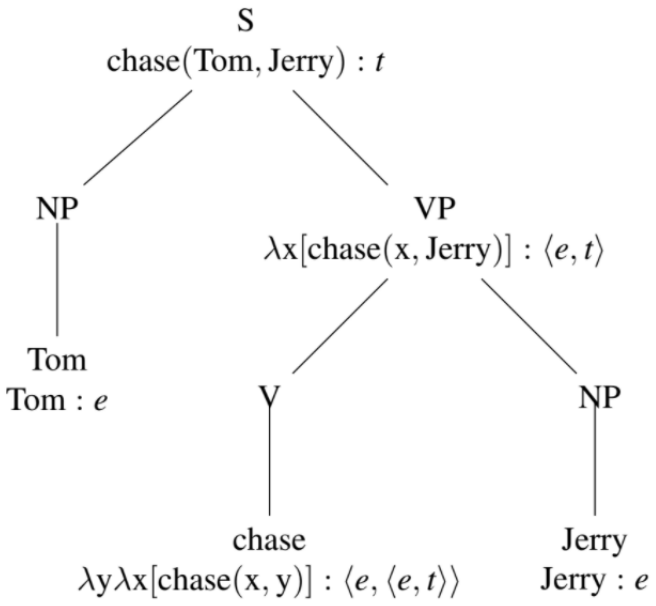

Figure 3.1: Derivation of the logical form of a sentence according to the Montague semantics.

The types defined by Montague over the functions and entities need to express some constraints that are traditionally called *selectional restrictions* (Katz and Fodor, 1963; Chomsky, 1965). For each typed function, there exist some selectional constraints that determine which are the appropriate arguments of a predicate. Variables of predicates can be typed with symbols that identify some semantic categories or features: a semantic composition is congruous if and only if the type of argument exactly matches the one required by the predicate. For example, the predicate *chase* require two arguments of type *ANIMATE*. It means that the sentence *Tom chases Jerry* satisfies both syntactic and semantic constraints. On the contrary, the sentence *The beer chases Jerry* does not, since the argument *beer* applied to the function *chase* is not of the type *ANIMATE* but *LIQUID*. So, according to this theory, in order to classify a sentence as correct it is not just needed to be grammatical (in the sense that all the grammatical rules are satisfied), but it is necessary that the selectional restrictions of all the application functions are exactly satisfied.
However, the way predicative and argument lexical items interact is a controversial aspect, especially when we want to define whether a complex expression is well-formed just according to the application functions in that expression. There are a number of contexts in

language (like the syntactic or semantic polymorphism, the coercion or the metonymy and metaphor phenomena) that appear to contradict the claim that meaning can be modeled compositionally in language, or at least they contradict the hypothesis of the selectional restrictions defined just above here. This is way, a lot of linguists over the years developed different semantic theories of the natural language.

3.3 Challenges to the compositional theory

The spread of the compositional theory among linguists lead to a great attention towards it and its main principles (Harris; 1954). Along years, a number of different theories have been defined starting from one or more assumptions of the Fregean compositionality, following them or contradicting them. Even if the compositional theory achieved a great success, now we can easily find a lot of examples of the natural languages against those assumptions, such as metaphor, metonymy and logical metonymy. In this subsection, we briefly discuss about the main challenges to the compositional theory and what is instead considered to be true nowadays.

Firstly, one of the main point of the compositional theory is that the meaning of words is context independent. The way words are used inside the sentence does not change the meaning associated to them. According to Frege, the meaning of each word is always the same; the conjunction of meaning of all the words of the sentence compose the meaning of the sentence itself. However, there are a lot of cases in which this assumption does not hold. This is confirmed, for example, by the different possible usages of the word *park*: it can be used as a verb (to indicate to bring a vehicle to a halt and leave it temporarily), or as a noun (to indicate a large public garden or an area devoted to a specified purpose). So, it is an evidence in favor of the hypothesis that the meaning of words is context-sensitive. Semantic composition does not always consist in the mere application of a function to an argument but even the predicate itself can be modified by the information brought by the argument. This phenomenon was called *co-compositionality* by Pustejovsky (1995). This is well modeled by some modern established natural language models such as the ones described in chapter 2, which extract the word representations according to the way they

occur in the text.

As already briefly told in section 3.2, another controversial point to the claim that meaning can be modeled compositionally in language is about the existence of the semantic constraints (called *selectional restrictions*) of predicates of complex expressions. It has been demonstrated that those semantic constraints should be regarded not as strictly binding but as selectional preferences (Wilks; 1978). Selectional preferences are the tendency for a word to semantically select or constrain which other words may (and not need to) appear in a direct syntactic relation with it. This selectional preferences should not be expressed in a binary term (allowed vs not-allowed). To support that view, the failure of those constraints does not always lead to a semantic anomaly because there can be some mechanism of coercion that still make the sentence reasonable (Pustejovsky, 1995; Asher, 2011; Lauwers and Willems, 2011). The coercion allows to add information that is not explicitly expressed in the sentence itself but depends on some contextual knowledge. So, the same predicate can have different selectional preferences: the functional application of a specific type of argument to the same predicate can adjust the meaning of it.

In order to explain such cases contradicting the compositional theory of Frege, Jackendoff proposed the so-called *enriched composition* (Jackendoff; 1997). According to Jackendoff, the semantic representation of a sentence may contain other information that are not expressed lexically. Those information must be present either in order to achieve well-formedness in the composition of the semantic representations or in order to satisfy the pragmatics of the discourse. The way the semantic representations are obtained is determined in part by the syntactic composition of the lexical items and partly by the internal structure of the representations themselves. Also the the principle of a perfect alignment between syntax and semantics is here rejected: the meaning of sentences are constructed from the meaning of their words plus some independent principles, some of which are correlated with the syntactic structure of the sentence while some other do not. In the dataset that we built we want to highlight all these phenomena.

4. Dataset

In this chapter it will be illustrated how the dataset for this thesis was built and it will be specified the reasons behind the implementation choices. We produced 1200 pairs of sentences. Each one was provided with a pair of labels one indicating whether the premise entails the hypothesis and the other specifying the type of the linguistic phenomenon taken under consideration for the inference task. As stated above, the way this dataset was constructed is deeply rooted in NLI tasks. The main goal of this work is to verify how the current state of the art models perform on this kind of task, with particular attention to the predicate-argument structure of texts, and what kind of information are effectively stored on the sentence representations these models produce.

4.1 Development

In order to develop the dataset used for our analyses, we have taken under consideration some already existing datasets for Natural Language Inference tasks such as the ones described in section 2.6. More specifically, the SuperGlue benchmark dataset for models evaluation have been taken as a gold for what regard its structure. The SuperGlue dataset consists of several hundred sentence pairs labeled with their entailment relations and tagged with a set of linguistic phenomena involved in that kind of relation. In order to to make the dataset suitable for analyzing many levels of natural language understanding, they identify four broad categories of linguistic phenomena. Those main categories of phenomena regards different kinds of linguistic information. In a pair of sentence more than one phenomena may be involved. The dataset was created manually, starting from some premise texts taken from different sources. The dataset is freely distributed in a tabular format with the following columns:

- *Lexical semantics*: it may contains the fine-grained category (or categories) centered on aspects of the word meaning.

- *Predicate-argument structure*: it may contains the fine-grained category (or categories) about the way words are combined into the whole sentence.
- *Logic*: it may contains a specific logical phenomenon (or phenomena) such as negation, conditional, quantification, monotonicity etc.
- *Knowledge & Common sense*: it may contains information for disambiguating word sense, syntactic structures, anaphora and more.
- *Domain*: the domain of the premise text, taken from the range: {News, Reddit, Wikipedia, ACL or Artificial}.
- *Premise*: the first text of the pair.
- *Hypothesis*: the second text of the pair.
- *Label*: the entailment label that can be *entailment* or *not_entailment*.

For our purposes, we decided to make some simplification in such structure since we focus on just one aspect of the pairs that is their predicate-argument structure. We dropped the columns *Lexical semantics*, *Logic* and *Knowledge & Common sense* since the respective information are not relevant in our case. So, the resulting schema of the dataset is composed by the columns:

- ***Predicate-argument structure***
- ***Domain***
- ***Premise***
- ***Hypothesis***
- ***Label***

We developed the dataset keeping it balanced on the *label* and *predicate-argument structure* columns. The column *domain* can have just two possible values: ARTIFICIAL and NEWS. The majority of the sentences were built artificially such that we could control

their complexity and the linguistic phenomena involved in the process of inference. Just few premises were taken from online newspapers in order to make some comparisons with the hand-crafted texts. As the benchmark dataset described in this work is used to evaluate models already trained on tasks of natural language inference, we decide to try controlling the complexity of the sentences.

Moreover, we reckon it more useful to consider simple pair of sentences, in such a way that just a linguistic phenomenon at a time is involved in the entailment relation: this practice is usually used and considered to be particularly efficient in many probing tasks. This is one of the differences with the diagnostic dataset of Glue, where there can be pairs of sentences in which more than one linguistic phenomenon is involved at the same time in their inference relation.

All the labels were assigned to each pair of sentences manually. The possible labels are: ENTAILMENT (if the hypothesis is entailed by the premise text), NEUTRAL (if there is not an entailment or not_entailment relation between the premise and the hypothesis) and CONTRADICTION (if the hypothesis contradicts the premise).

In 4.1 we illustrate an example of the dataset.

Predicate-argument structure	**Domain**	**Premise**	**Hypothesis**	**Label**
Core args	Artificial	They eat a cake.	They eat.	Entailment
Datives	Artificial	I gave her the keys back.	I have her keys.	Contradiction

Table 4.1: Example of two rows of the dataset built

4.2 Categories

Since the category of the predicate-argument structure is too vague, we decided to identify some finer categories in order to point out which are the specific cases the models are able to correctly recognize and which do not. Again, we decided to take into account the same

fine-grained categories used in Glue and SuperGlue (Wang et al; 2019), that are going to be described in details in this sub-section. These categories emphasize both syntactic and semantic aspects, going from syntactic ambiguity to semantic roles and linguistic references, that we reckon to be crucial for understanding the entailment relationships.

4.2.1 Core arguments

A core argument of a verb is a subject, a direct object or an indirect object. In other words, a core argument is a noun phrase that fulfills semantic roles determined by a verb, or more generally by a predicate. Verbs are associated with a set of semantic participants that somehow take part in the event described. Some of the verb's semantic participants can be mapped to roles that are syntactically relevant in the clause, like a subject or a direct object: this specific case of semantic participants marks what core arguments are and it distinguishes them from oblique arguments. Depending on the language, the verb may also specify requirements on the position of the individual arguments and on their form, such as morphological case marking or preposition. This definition of core arguments is consistent with the *Role and reference grammar* (RRG) developed by Foley and Valin in the 1980s and which incorporates many points of view of current functional grammar theories.

While drawing the pairs of sentences, we try to represent all the possible cases, or at least the most frequent ones in English. For example, we tried to investigate the entailment relations between pairs of sentences in which some core arguments have been reversed, substituted, deleted and/or added. Below we report some examples of pairs of sentences in which the linguistic phenomenon involved in the inference relation between the premise and the hypothesis regards the core arguments. On each phrase we emphasize in italic the specific core argument (or the core arguments) that is involved in that relation.

- **Premise**: They eat *a cake*.
  Hypothesis: They eat.

- **Premise**: She left.
  Hypothesis: She left *the house*.

- **Premise**: I have not *a yellow hat*.
  Hypothesis: I have not *a yellow skirt*.

- **Premise**: *Luke* saw *the dog*.
  Hypothesis: *The dog* saw *Luke*.

- **Premise**: *John* met *Rachel at the school*.
  Hypothesis: *Rachel* didn't meet *John*.

4.2.2 Prepositional phrases

A prepositional phrase is a syntactic category included in the one of the adpositional phrases. An adpositional phrase is characterized by the presence of an adposition (that can be a preposition, postposition or circumposition) as head and usually a complement such as a noun phrase. A prepositional phrase is a phrase containing a preposition, its object and any words that modify the object. Most of the time, the object that the prepositional phrase modifies is a verb or a noun: the first case is usually named *adverbial phrase*, while the second one *adjectival phrase*. Some of the most common english prepositions that begin prepositional phrases are *to*, *of*, *about*, *at*, *for*, *in*, *with*, *before*, *after* and *during*. Below we report some examples of pairs of sentences highlighting this type of phenomena that have been inserted into this dataset.

- **Premise**: *Before the lunch* we will meet at the gym.
  Hypothesis: We will meet at the gym.

- **Premise**: I will arrive *at New York tomorrow*.
  Hypothesis: I am at New York.

- **Premise**: *Marc* ran *into Josh on his way to work*.
  Hypothesis: *Josh* ran *into Marc*.

- **Premise**: That men *with the blue t-shirt* is the gardener.
  Hypothesis: That men *wearing a blue t-shirt* is the gardener.

- **Premise**: Our team won *against all odds*.
  Hypothesis: Our team won *against their team*.

- **Premise**: Cities *across the country* are bracing *for protests regardless of the verdict*.
  Hypothesis: Cities *across the country* are bracing.

This kind of phenomenon have been considered here since the prepositional phrase attachment is a particularly difficult problem that most syntactic parser in Natural Language Processing systems continue to struggle with. It is both a syntactic and a semantic problem, since prepositional phrases can be very complex and express a wide variety of semantic roles. Moreover, prepositional phrases often semantically apply beyond their direct syntactic attachment.

4.2.3 Nominalization

The nominalization (also known as ”nouning”) is the use of a word which is not a noun as it would be, or as the head of a noun phrase. This process is sometimes due to the fact that people tend to isolate some activities as abstract conceptual units in their minds, so it lead to represent them as nouns (for example, *analyzing* can become *analysis* or *measuring* can become *measurement*). The words involved in this kind of linguistic phenomenon are usually a verb, an adjective, a gerundive or an adverb. This change in the functional category can cause also a morphological transformation. In English there are essentially two types of nominalization: creating a noun trough the addition of a derivational affix to the original word, or the same word can be used as a noun without any transformation in its morphology. An example of the latter one can be done with the word *change*. Let's consider the phrase *I change.* where the word acts like a verb, and the phrase *I need a change.* where the same word is used as a noun. Instead, when a nominalization happens with a morphological transformation there are some common suffixes added to words depending on their types. Some of these suffixes are reported in table 4.2.

Nominalized type	Suffix	Example
Nominalized adjective	-bility	capability
	-ity	stupidity
	-ness	happiness
Nominalized verb	-dom	freedom
	-edge	knowledge
	-ion	reaction
	-ment	government
Nominalized gerundive	-ing	writing

Table 4.2: Example of some of the most common morphological transformations in nominalizations

Nominalized sentences tend to insert much of their information into the subject position, which may hinders readability and make them more difficult to be understood. A nominalization can mask the key verb of the sentence and hence, we often risk to lose important information even if it not always happens.

- **Premise**: Laura *made her apologies* for not having told him the truth.
  Hypothesis: Laura *apologized* for not having told him the truth.

- **Premise**: Lisa collaborated in *the publication of* the novel.
  Hypothesis: Lisa collaborated in *publishing* the novel.

- **Premise**: It was an unacceptable *discrimination* by their side.
  Hypothesis: *Discriminating* is unacceptable for them.

- **Premise**: There was no *interference* during their radio program.
  Hypothesis: Something *interfered* during their radio program.

- **Premise**: Their *disagreement* was clear to everyone.
  Hypothesis: They *disagree*.

4.2.4 Genitives and partitives

The genitive case in English grammar is the case that is used for a noun, pronoun, or adjective that modifies another noun. The genitive case is most commonly used to show possession, but it can also show a thing's source or a characteristic/trait of something. It can be easily recognized because it is typically composed by adding an apostrophe followed by a "s" to the end of a singular noun, or just adding an apostrophe to a plural noun already ending by -s.
The partitive case is a case that expresses the partial nature of the referent of the noun it marks, as opposed to expressing the whole unit or class of which the referent is a part. The partitive case can be found in existential clauses; in clauses where nouns are accompanied by numerals or units of measure; or in predications of materials from which something is made. In general, both the genitive and the partitive case are quite easy to be recognized in english texts. Below we report some examples.

- **Premise**: This is *the notebook of my cousin Albert*.
  Hypothesis: This is *my cousin Albert's notebook*.

- **Premise**: Taylor is *a guy of many talents*.
  Hypothesis: Taylor has many talents.

- **Premise**: Someone stole *George's bicycle* that was left at the corner of the street.
  Hypothesis: *The bicycle of George* is at the corner of the street.

- **Premise**: *The administrator's intention* is making a new contract.
  Hypothesis: *The intention of the administrator* is to make a new contract.

- **Premise**: Jake is *Charles's brother*.
  Hypothesis: Charles is *the brother of Jake*.

4.2.5 Datives

The dative case is a grammatical case typical of nouns and pronouns that mark their relationship to other words in the sentence. It is used to indicate the recipient or beneficiary

of an action. In English, the dative case marks the indirect object of a verb. An indirect object is the recipient of a direct object which is the accusative case. As for genitives and partitives, the dative case is quite easy to be recognized in English texts because it usually follows specific patterns. Here are some examples extracted from the dataset built for this work.

- **Premise**: I gave *her* the keys back.
  Hypothesis: I gave the keys back *to her*.
- **Premise**: Anne passed *her* the ball.
  Hypothesis: Anne passed the ball.
- **Premise**: Marc took *his cat* to the vet.
  Hypothesis: Marc was to the vet with his cat.
- **Premise**: *You* should tell *your mum* the truth.
  Hypothesis: *Your mum* should tell *you* the truth.
- **Premise**: Barney will send *him* the presentation tomorrow.
  Hypothesis: Barney will send the presentation tomorrow.

4.2.6 Active and passive

Tenses can have an active form or a passive form, so sentences can be either active or passive. Active sentences are characterized by the fact that the thing doing the action is the subject of the sentence, while the thing receiving the action is the object. In English, most of the sentences of both written and speech texts are active: it is the standard form. On the contrary, passive sentences are characterized by the fact that the thing receiving the action is the subject of the sentence, while the thing doing the action can be included in the sentence or not. The thing doing the action is included in a passive form when we want to emphasize its role. When it happens, the thing doing the action is introduced in the sentence by the proposition *by*. A tense can be used in a passive form also when it is not known the person (or the thing) doing the action that is described.

The difference between active and passive sentences is usually quite easy to be automatically recognized due to the fact that the have different construction mechanism and also active and passive forms have usually a specific pattern. This can be seen also looking at the following examples inserted in the dataset and classified with different labels (not reported here).

- **Premise**: I bought the sunglasses in his shop.
  Hypothesis: The sunglasses were bought by me in his shop.
- **Premise**: My dad borrowed a camper for summer holidays.
  Hypothesis: The camper was borrowed for holidays.
- **Premise**: Gruenig was questioned by activists at the press conference.
  Hypothesis: Activists questioned Gruenig at the press conference.
- **Premise**: Gruenig was questioned by activists at the press conference.
  Hypothesis: Gruening asked a question to activists.
- **Premise**: Messi had not scored any goal at the last game.
  Hypothesis: A goal has been scored the last game.

4.2.7 Relative clauses

A relative clause is a kind of dependent clause, so a clause that cannot function syntactically as a complete sentence by itself but has a nominal, adjective or adverbial function within another sentence. The sentence to whom it depends is called *independent clause* or *main clause*. In particular, the relative clause functions like an adjective, giving more information about a noun of the main clause. A relative clause can be easily recognized as it is always introduced by a relative pronoun, which substitutes for a noun, a noun phrase or a pronoun when the sentences are combined. The relative pronouns for English are reported in table 4.3.

Pronoun	Usage	Meaning
who	animate	substitutes for subject nouns or pronouns (e.g. he, she, they)
whom	animate	substitutes for object nouns or pronouns (e.g. him, her, them)
whose	animate, inanimate	substitutes for possessive nouns or pronouns (e.g. his, hers, theirs)
that	animate, inanimate	used only in restrictive relative clauses, for either subject or object
which	inanimate	can be used in both restrictive and non-restrictive relative clauses, for either subject or object

Table 4.3: English relative pronouns.

Relative clauses can be *defining* or *non-defining*. A defining relative clause tells which noun we are talking about. It adds essential information about someone or something (e.g. in the sentence *I like the boy who lives next door.*, the relative clause is essential to identify which boy I am talking about). A defining relative clause usually comes immediately after the noun it describes. Instead, a non-defining relative clause adds extra information about something even if those information are not necessary to understand the whole sentence (e.g. in the text *I live in Milan, which has some fantastic museums*, the expression"which has some fantastic museums" adds information that are not necessary to identify Milan). A non-defining relative clause is usually enclosed within commas, but not always this is true.
Of course, relative clauses can sometimes make the text a lot more complex. This causes a great amount of ambiguity in English. However, in this dataset we try to control their complexity, as it can be seen by the following examples.

- **Premise**: He likes driving cars *that are fast*.
  Hypothesis: He likes driving cars.

- **Premise**: The secretary *who she is looking for* must speak at least two languages.
  Hypothesis: The secretary must speak more than one language.

- **Premise**: The music *which Jiulie listen to* is good.
  Hypothesis: Jiulie listens to good music.

- **Premise**: The club *where Mary used to go last year*, just closed.
  Hypothesis: The club just closed.

- **Premise**: The man you met yesterday is the project manager.
  Hypothesis: The man *that you met yesterday* is the project manager.

4.2.8 Restrictivity

Restrictivity is most often used to refer to a property of uses of noun modifiers. In semantics, a modifier is said to be restrictive if it restricts the reference of its head. For example, in the sentence *I like the red t-shirt more than the blue one.*, *red* and *blue* are restrictive adjectives because they allow us to restrict which t-shirt we are referring to. Restrictive modifiers are sometimes also called *defining*, *essential* or *necessary*. On the contrary, in the sentence *What's an amazing view.* the adjective *amazing* it is not restrictive because it is not necessary to the identification of the noun but it just adds some extra information. Non-restrictive modifiers can also called *non-defining*, *descriptive* or *unnecessary*. Usually, modifiers that are commonly used with a non-restrictive sense are appositives, relative clauses starting with the prepositions *which* or *who*, and expletives.
English does not generally marks modifiers for restrictiveness. However, in certain cases, while restrictiveness can be marked syntactically though the lack of commas, restrictive modifiers are integrated. In speech this difference can be marked through the intonation, making a pause. Furthermore, although restrictive clauses can be headed by any of the relative pronouns (already reported in table 4.3), non-restrictive clauses can, at least, be headed only by the pronouns *who* or *which*. Eventually, restrictive noun modifiers can sometimes be marked periphrastically, incorporating them into relative clause (e.g. *John's cousin, who lives in the United States, is a professional baseball player.*).
The difference in the restrictiveness semantics of noun modifiers play an important role in natural language understanding tasks in general. This distinction is often highlighted in entailment relations between sentences. However, there can be cases in which the differ-

ence between a restrictive and a non-restrictive noun modifier can be understood just with some contextual knowledge. Below we report some examples of different usages of noun modifiers according to a restrictiveness point of view.

- **Premise**: A couple *holding hands* walks down a street.
  Hypothesis: A couple walks down a street.
- **Premise**: Karl and Lewis are *travel lovers*.
  Hypothesis: Karl and Lewis are lovers.
- **Premise**: Karl and Lewis are *travel lovers*.
  Hypothesis: Karl and Lewis are *sport lovers*.
- **Premise**: The guy *living in Baker Street 6* is a friend of mine since the school time.
  Hypothesis: The guy is a friend of mine since the school time.
- **Premise**: The year *that just ended* was full of new scientific discoveries.
  Hypothesis: *That* year was full of new scientific discoveries.
- **Premise**: The year *that just ended* was full of new scientific discoveries.
  Hypothesis: *2020* was full of new scientific discoveries.

4.2.9 Ellipsis and implicits

The ellipsis is a linguistic phenomenon that can regard syntactic, semantic and pragmatic aspects. Ellipsis consists of leaving out one or more items which are normally expected to be used in a sentence if the relative grammar rules are followed. The item that was left out must be supplied by the listener or the reader in order to understand the sentence. Often, the argument of a verb or other predicates are the words omitted in the text, with the reader (or the listener) filling in the gap. In writing, ellipsis can be marked with the three dots to show the point on the sentence in which the word misses. The dots can also be used to indicate a long pause or a speech trailing off. In speech, especially in informal speeches, people often leave out unnecessary information and speak in shorthand. It is a way to be brief and not repetitive, but still clear in the communication. Here are some pairs of sentences of our dataset, belonging to this fine-grained category.

- **Premise**: Marc can play the piano, his sister the violin.
  Hypothesis: Marc can play the piano, his sister can *play* the violin.

- **Premise**: Marc can play the piano, his sister *likes* the violin.
  Hypothesis: Marc can play the piano, his sister the violin.

- **Premise**: I get a B that was the best of the class.
  Hypothesis: I get a B that was the best grade of the class.

- **Premise**: I have Susan's book, you have Bill's.
  Hypothesis: You have *Bill's book*.

- **Premise**: The more they stay in the house, the harder it will be to leave.
  Hypothesis: If they stay more in the house, it will be harder for the them to leave *the house*.

4.2.10 Anaphora and coreference

The coreference is a linguistic phenomenon that occurs when two or more expressions in a text refer to the same referent that can be either a person or a thing. Its resolution is a well-studied problem for both semantic and syntactic points of view. Coreference is the main concept underlying binding phenomena in the field of syntax. The binding theory is the component of grammar that regulates the interpretation of noun phrases. The task of the binding theory is to determine which noun phrases in a given syntactic domain can be coreferential, and eventually explore the syntactic relationship that exists between the coreferential expressions in sentences and texts. There are distinct structural conditions that determine the binding possibilities for noun phrases. When two expressions are coreferential, one of them is usually a full noun phrase (the antecedent) and the other(s) is (or are) an abbreviated form (like a proform or an anaphor).
There are numerous kinds of different coreference examples, such as anaphora, cataphora, split antecedents, etc. When the second coreferential item is a proform, we can have an anaphora or a cataphora. The anaphora is the use of an expression whose interpretation depends upon another expression in context, that is its antecedent. The anaphoric term is

called an anaphor. The anaphora phenomenon, that is the most frequent one in English, is exactly the opposite of the cataphora. The cataphora consists of the use of an expression that depends upon a postcedent expression.
Some of the pairs of sentences of the dataset that highlight the coreference phenomenon are the following ones.

- **Premise**: John has a red *car* while his girlfriend has a black *one*.
  Hypothesis: John's girlfriend has a *car*.

- **Premise**: If my son moves *to Florida*, I will move *too*.
  Hypothesis: I will move *to Florida*.

- **Premise**: The organisation also lists *online threats*, such as *harassment, trolling and state surveillance*, as undermining journalists' work across the continent.
  Hypothesis: The organisation also lists *harassment, trolling and state surveillance*, as undermining journalists' work across the continent, even in countries where freedom is held in high regard.

- **Premise**: The young girl bought *a skirt* at the market in the city center, but she had to return *it* because *it* was too big.
  Hypothesis: *The skirt* is too big.

- **Premise**: Marc told her *not to go to on vacation* but then he regretted *it*.
  Hypothesis: Marc regretted *to have told her not to go on vacation*.

- **Premise**: Marc told her *not to go to on vacation* but then he regretted *it*.
  Hypothesis: Marc regretted *to not have gone on vacation*.

4.2.11 Intersectivity

An intersective modifier is an expression which modifies another by delivering the intersection of their denotations. In other words, when a modifier is intersective, its contribution to the truth conditions of the sentence does not depend on the particular expression it modifies. This means that one can test whether a modifier is intersective by seeing whether

it gives rise to valid reasoning patterns. Intersectivity is a crucial point for natural language inference tasks since the fact that a modifier is intersective can drive the meaning of the whole sentence. However, modifiers can be ambiguous because the same modifier can have both intersective and non-intersective interpretations. For examples, in the sentence *Olive is a beautiful dancer* the modifier *beautiful* can be seen as a intersective one as well as a non-intersective. Intersectivity is often related to factivity.

- **Premise**: If they feel well, they will *go to the beach* and *organize a party at Susan's house*.
  Hypothesis: If they feel well, they will *organize a party at Susan's house*.

- **Premise**: If they feel well, they will *go to the beach* and *organize a party at Susan's house*.
  Hypothesis: If they feel well, they will *go to the beach*.

- **Premise**: *Families* and *players* alike ran for cover, some hiding in the dugouts, some rushing for the exits or nearby buildings.
  Hypothesis: *Players* ran for cover, some hiding in the dugouts, some rushing for the exits or nearby buildings.

- **Premise**: The autopsy found no evidence *of an allergic reaction to chemicals*, *nor of internal or external injuries*, Dr Francisco Diaz said.
  Hypothesis: The autopsy found no evidence *of internal or external injuries*, Dr Francisco Diaz said.

- **Premise**: The autopsy found no evidence *of an allergic reaction to chemicals*, *nor of internal or external injuries*, Dr Francisco Diaz said.
  Hypothesis: The autopsy found no evidence *of an allergic reaction to chemicals*, Dr Francisco Diaz said.

4.2.12 Coordination scope

The coordination scope is a fine-grained category that deal purely with resolving syntactic ambiguities. The coordination is a complex syntactic structure that links together two or

more elements which are called *conjuncts*. The conjuncts are combined into a larger unit and still have the same semantic relations with other surrounding elements. The presence of a coordination is often highlighted by the presence of a coordinating conjunction (e.g. *and*, *or*, *but*). The coordinator(s) and the conjuncts together form an instance of coordination called *coordinate structure*. The coordinate structures are divided into two main categories: coordination and subordination. This sub-section (and so this fine-grained category of the dataset) deals with both of them.

Coordination is one of the most studied fields in theoretical syntax, but despite decades of intensive examination and due to the complexity of the field, theoretical accounts differ significantly and there is no consensus on what is the best way of analyzing this kind of structures. Moreover, the semantic of certain coordination structures can also be a bit controversial due to the different possible ways of interpreting them. Syntactic parsers can find difficult to analyze coordination structures due to the constraints over the coordination scopes. The notion of scope in natural language is the same as in logic: the scope of an operator is that part of the expression (or text) on which it performs its characteristic action. If one operator is within the scope of another, their relative scope determines their order of operations. In English, the coordinators can be either within the scope of prepositions (e.g. *I bought a present for [John and Marvin]*), or outside their scope (e.g. *I bought a present [for John] and [for Marvin]*). Perhaps, there can be a slight difference on the semantics according to the different case: in the first sentence reported as example, we suppose that just a present was bought and John and Marvin will share it, while in the second example we suppose that a present was bought for John and another one for Marvin.

Here we report some of the pairs of sentences of the dataset following under the category of *coordination scope*.

- **Premise**: He *cooked some pasta* and *ate with his friends*.
  Hypothesis: He *ate with his friends*.

- **Premise**: He *cooked some pasta* and *ate with his friends*.
  Hypothesis: He *ate some pasta with his friend*s.

- **Premise**: Samuel planned to go to *Florence and Pisa* the next week.

Hypothesis: Samuel planned to go to *Pisa* the next week.

- **Premise**: They published the scores they achieved *but* didn't released any information about their strategy.
  Hypothesis: They published the scores they achieved *and* didn't released any information about their strategy.

- **Premise**: Lewis has to stay home *not because he is sick*, *but rather to look after his little brother*.
  Hypothesis: Lewis has to stay home *not because he is sick or to look after his little brother*.

- **Premise**: Lewis has to stay home *not because he is sick*, *but rather to look after his little brother*.
  Hypothesis: Lewis has to stay home *not because he has to look after his little brother*.

4.3 Data distribution

In order to guarantee a good representation of all the linguistic phenomena regarding the predicate-argument structure of sentences we decided to have a balanced number of pairs of sentences falling under each specific fine-grained category. So, for each fine-grained category there are 100 pairs of sentences in the dataset. Since we defined 12 categories, the dataset is composed of 1200 labeled pairs of sentences.
Moreover, we tried to keep the dataset balanced also on the *label* column, that is the one indicating whether the type of inference relation between the premise and the hypothesis is of *entailment*, *contradiction* or *neutral*. However, this was no such a easy job so the labels have a quite different distribution over the dataset. In figure 4.1 are reported the percentage distribution of the labels.

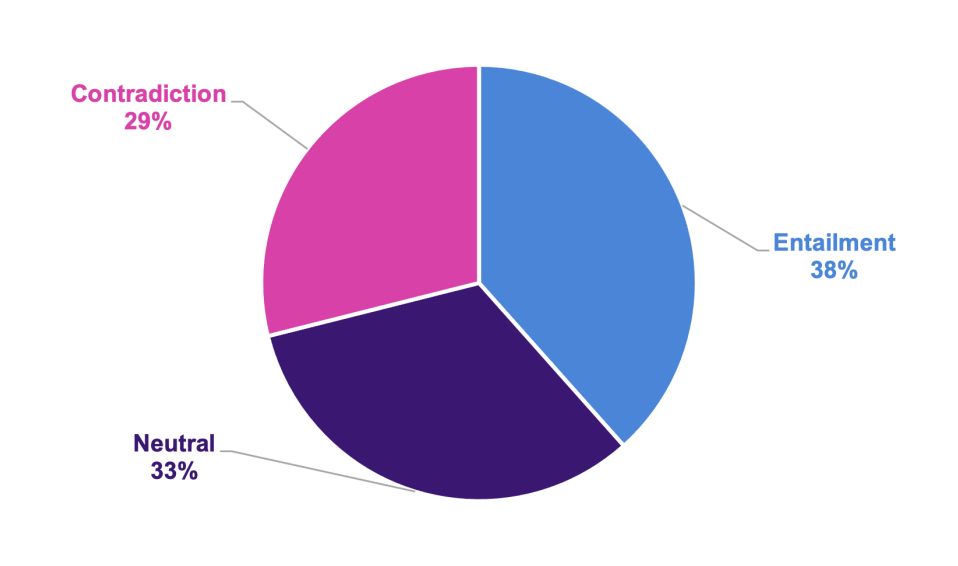

Figure 4.1: Distribution of label on the dataset

The most common label over the whole dataset is the *entailment*. There are 461 pairs of sentences labeled with an *entailment* inference relation, that is the 38% of the dataset. Then, the 33% of labels of the pairs is equal to *neutral*, that means it occurs 392 times. The remaining 29% of the cases (347 occurrences) have a label *contradiction* . So, even if the occurrences of the different labels are not the same, there is not such an enormous difference between them.

The percentage distribution of labels in the dataset have been counted also depending on the particular phenomenon of the predicate-argument structure. Here there is a greater variance along the macro categories. Their percentage distribution is shown in figure 4.2

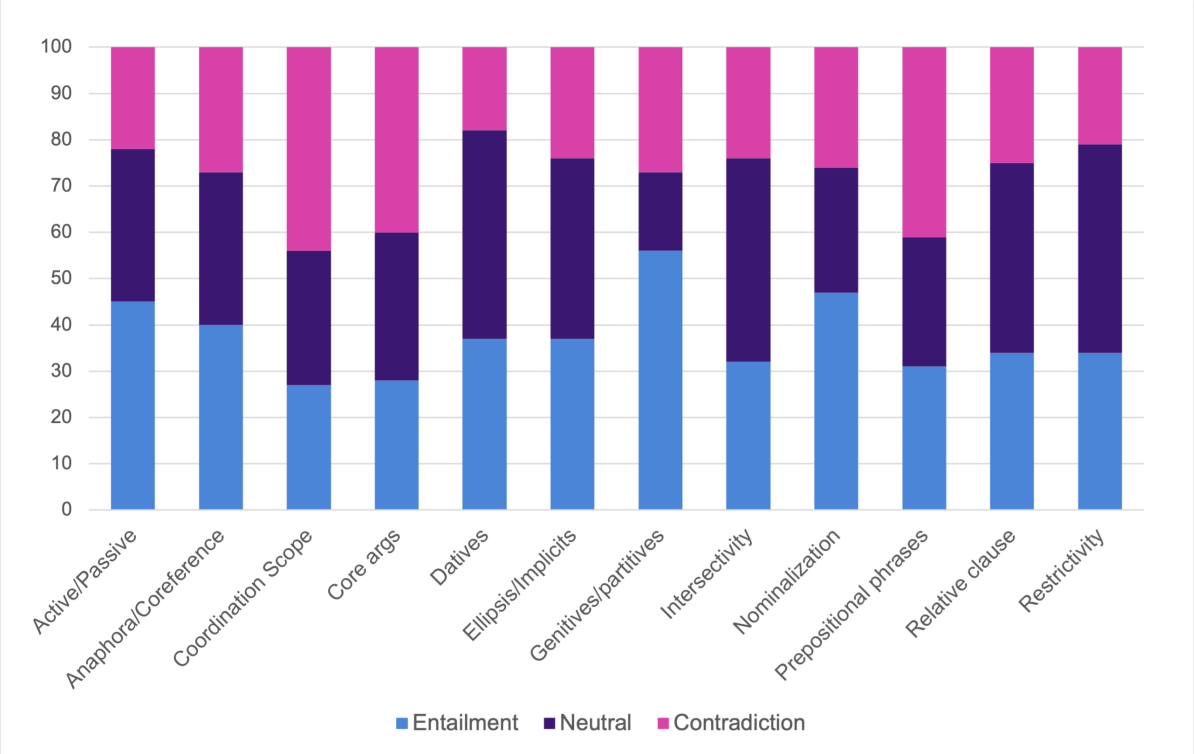

Figure 4.2: Distribution of labels per category

5. Experiments

The dataset built as described in chapter 4 was then used to conduct some experiments. The main goal of such experiments was to check how much information some state-of-the-art neural network models are able to grasp about the entailment relationship between pairs of sentences. The models used are all pre-trained and freely available on the Internet. Furthermore, the results obtained on our dataset were compared with the ones obtained by the same models on the dataset the were trained on for solving a Natural Language Inference task.
In this chapter we will analyze the models selected to conduct the experiments, their setups and the results obtained.

5.1 State-of-the-art models

The very first stage of the experimental phase was to decide which kind of experiments we wanted to conduct and what models we wanted to use. In order to make such a choice, we analyzed some of the best models used nowadays for solving Natural Language Inference tasks. Some of the greatest achievements on this field were made by using the corpora we described in section 2.6. Just for citing some of them, some state-of-the art models were trained on the SNLI or the MultiNLI corpus and achieved an accuracy higher than 90%. In the web-pages of these two corpora were reported the best performing models and their results: here we have a brief look at some of them.
Liu et al. (2019) presented a Multi-Task Deep Neural Network (MT-DNN) for learning representations across multiple Natural Language Understanding tasks that achieved the following test accuracy values: 91.6% over the SNLI corpus, 87.9% over the in-genre examples from MultiNLI, 87.4% over the cross-genre examples from MultiNLI, and 82.7% on the GLUE benchmark. The architecture of the MT-DNN is composed of multiple layers, some of which are shared across all tasks while others represent task-specific outputs. The knowledge distillation method (Hinton et al.; 2015) in the multi-task learning setting

is applied on the MT-DNN. Using the SNLI and SciTail datasets, they also demonstrated that the model allows domain adaptation with substantially fewer in-domain labels than the pre-trained BERT representations [13].

Another model that performed well enough both on MultiNLI and SNLI is RoBERTa (Liu et al.; 2019). RoBERTa is a replication study of BERT that carefully measures the impact of many key hyper-parameters and of the size of the training data. Moreover, with RoBERTa the next sentence prediction objective was removed and dynamic changes over the masking pattern were applied to the training data. Liu et al. obtained the 90.2% of accuracy on the MultiNLI cross-genre data, against the 89.6% obtained by BERT over the same data [14].

A slightly higher test accuracy (91.9%) on the SNLI corpus was obtained by Zahng et al. (2019) by using the Semantics-aware BERT model (also named SemBERT), which is a Transformer model capable of incorporating explicit contextual semantics from pre-trained semantic role labeling over a BERT backbone. SemBERT is as simple in concept as BERT but more powerful [23].

Finally, the best model performing on the SNLI up to now is the so-called Conditionally Adaptive Multi-Task Learning (CA-MTL) [41]. CA-MTL is a novel Transformer based Adapter consisting of a new conditional attention mechanism as well as a set of task-conditioned modules that facilitate weights sharing. Moreover, they used a multi-task data sampling strategy to mitigate the negative effects of data imbalance across tasks. The model proposed by Pilault et al. is composed of a decoder that is specific for each task, while the input embedding layer and the lower Transformer layers are frozen. The upper Transformer layer and conditional alignment module are modulated with the task embedding. The test accuracy obtained with this model on the SNLI is 92.1% and 85.9% on the GLUE benchmark.

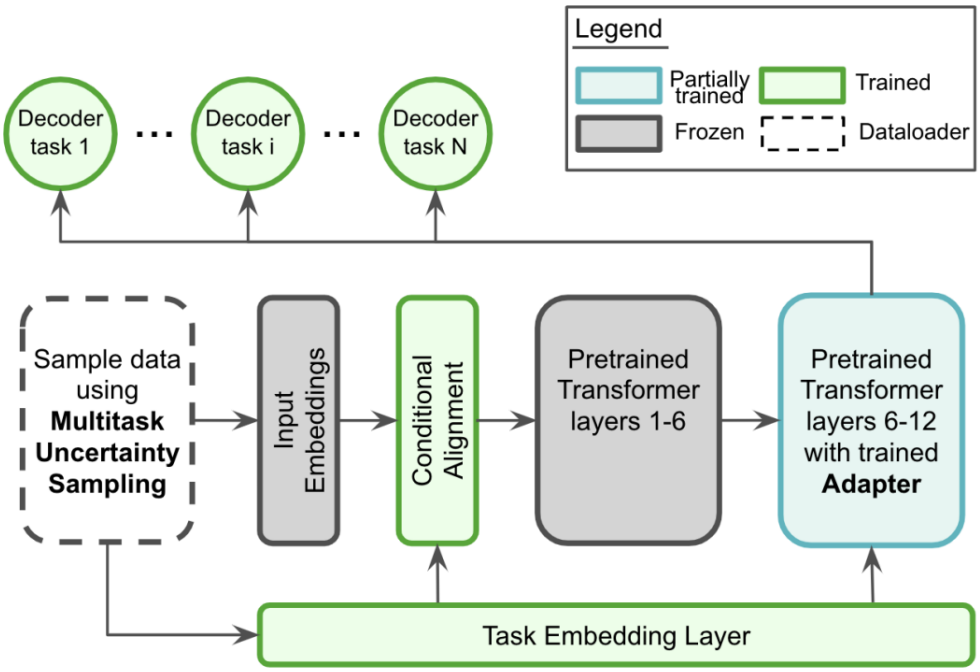

Figure 5.1: CA-MTL base architecture.

As demonstrated by these examples, the state-of-the-art models for NLI tasks are all Transformer based. Substantial improvements over previous state-of-the-art results of shared challenges or online benchmarks like GLUE have been obtained after the development of the well-known BERT approach (Devlin et al.; 2018). Indeed, a lot of participants and researchers' groups used BERT models with task specific fine-tuning: the formula "*BERT-variant* + *fine-tuning*" has been proved to be particularly efficient and it has continued to improve over time with newer works, constantly pushing the state-of-the-art forward on the GLUE benchmark. The best results over it can be seen on the GLUE leaderboard, where participants publish the results they obtained and details about the model and the approach they used.

5.2 Experiments setup

Given the astonishing success of Transformers and BERT-based models, we decided here to use some of them that are freely available on the huggingface website [2]. In particular, the main goal of our work was to analyze the quality of the representations learned by such models pre-trained for solving Natural Language Inference tasks. We used the models on our dataset such that we could focus on their ability to recognize the entailment relation between pairs of sentences in which phenomena that highlight the predicate-argument structure play a key role on the inference. We decided to select models that were pre-trained on the some corpus, the MultiNLI, such that a comparison between them could be as reasonable as possible. Those models are:

- *facebook/bart-large-mnli*
- *roberta-large-mnli*
- *microsoft/deberta-large-mnli*
- *microsoft/deberta-v2-xlarge-mnli*

For each model we selected the large version so that again an equal comparison would be made, with a single exception: we tried DeBERTa released by Microsoft in two different size (large and xlarge) in order to analyze its effect in terms of results obtained over our dataset. For each of the selected models, we performed a forward phase on the examples provided by the dataset we just created.

In order to provide a measure of how well the models performed, we have taken into account their respective *confusion matrices*. A confusion matrix (or *error matrix*) is an established technique for summarizing the performance of a classification algorithm. Each row of the matrix represents one actual class of the dataset, while each column of the matrix represents one class predicted by the classifier. So, the rows stands for the classes of the so-called *gold standard*, while the columns for the predicted classes [33]. In a binary

[2] `https://huggingface.co/`

classification scenario, the confusion matrix will have two rows and two columns as in the example in table 5.1.

	Positive	**Negative**
Positive	True positive	False negative
Negative	False positive	True negative

Table 5.1: Structure of a confusion matrix for a binary classification scenario.

The conjunction between *true positive* and *true negative* represents the total observations that have been correctly predicted by the model; while the conjunction between *false positive* and *false negative* represents the total number of misclassified elements. More precisely, the internal cells of the table 5.1 are defined as:

- **True Positive** (TP): a test result that correctly indicates the presence of a condition or characteristic;
- **True Negative** (TN): a test result that correctly indicates the absence of a condition or characteristic;
- **False Positive** (FP): a test result that indicates the presence of a condition or characteristic while it is actually absent (error in the classification);
- **False Negative** (TP): a test result that indicates the absence of a condition or characteristic while it is actually present (error in the classification);

So, the goal of a generic classifier is to maximize the sum of *true positive* and *true negative*, that means to minimize the sum of *false positive* and *false negative*. The structure of the table 5.1 can be adapted to a multi-classification scenario just be adding the columns and rows for all the possible classes. Starting from a confusion matrix, it is possible to compute others performance metrics which have been used also in our experiments: *precision*, *recall*, *accuracy* and *F1-Score*.

The ***accuracy*** is the fraction of predictions the classification model got right. For binary

classification, the accuracy is computed with the following formula.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

However, the accuracy does not tell the full story when the dataset is imbalanced on the possible classes (when there is a significant difference between the number of positive and negative examples). For imbalanced datasets, precision and recall are more appropriate performance metrics.

The ***precision*** measures what is the number of instances correctly predicted as positives over the total number of positive predictions. In other words, it tries to answer at the question "What proportion of positive identifications was actually correct?". It is computed as:

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

The ***recall*** (also known as *sensitivity*) is the fraction of the correctly positive classified instances over the total number of instances that are actually positive. It is computed with the following formula.

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

Ideally, we are usually interested in both precision and recall: this is the reason why the F1-Score was introduced. The ***F1-Score*** can be considered as an improvement of two simpler performance metrics, as long as it is the harmonic mean of precision and recall. An harmonic mean is an alternative metric for the more common arithmetic mean, which is particularly useful when computing an average rate, as in this case. The F1-Score formula is the following one.

$$F1Score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

So, for each Transformer model we performed a feed-forward phase and then we evaluated its performance on our dataset in terms of the metrics described in this section. The

results were firstly computed on the whole dataset. Than, in order to see how the specific fine-grained categories of the predicate-argument structure contributed to the final model performances, we filtered the dataset out by each of the category described in section 4.2. Finally, we compared the results obtained along the different models.

5.3 Bart

Bart is a denoising autoencoder for pretraining sequence-to-sequence models that was introduced by some researchers from Facebook (Lewis et al.; 2019) [12]. It uses a standard Transformer-based neural machine translation architecture which is a generalization of BERT, GPT and other recent pretraining schemes. The base model uses 6 layers in the encoder and decoder, while the larger version uses 12 layers in each. The differences in the architecture when compared this model to BERT is that (1) each layer of the decoder additionally performs cross-attention over the final hidden layer of the encoder; and (2) it does not uses an additional feed-forward neural network before word prediction like BERT does. Moreover, BART contains about 10% more parameters than the equivalent sized BERT.

BART pre-trains a model combining Bidirectional and Auto-Regressive Transformers. The pre-training phase is composed of two stages: firstly the text is corrupted with an arbitrary noising function; then a sequence-to-sequence model is learned to reconstruct the original text by computing a cross-entropy between the decoder's output and the original text. This ensures a noising flexibility: arbitrary transformations can be applied to the original text (including token masking, token deletion, text infilling, sentence permutation or document rotation). The extreme case of the possible text corruptions is when all the initial information got lost: in this case BART is equivalent to a language model.

The representation learned by BART can be used in many ways for downstream tasks, such as token classification, sequence generation and machine translation tasks. The tasks over which fine-tuning of BART resulted particularly effective are text generation and comprehension tasks.

For our experiments we used the larger version of the model BART freely released by Facebook after it has been pre-trained over the MultiNLI corpus. We performed a feed-forward phase on our dataset. More precisely, we gave it as input the pairs of premise-hypothesis we drew. For each pair of sentences the model assigned a probability value of belonging to each possible class (*entailment*, *neutral* and *contradiction*). The most probable class is the one predicted by the classifier. At the end, the predicted labels were compared to the actual ones.

Once the model predicted the most probable label for each pair of sentences, we plotted the resulting confusion matrix which is shown in figure 5.2.

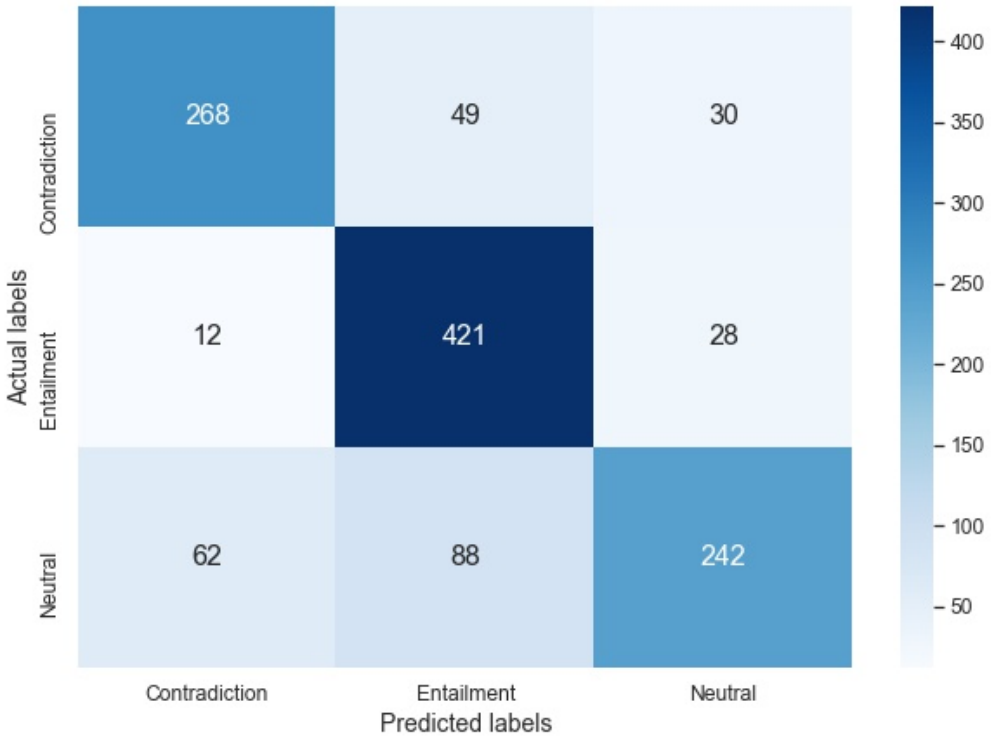

Figure 5.2: Confusion matrix for Bart Large.

As illustrated in section 4.3, the more frequent label in our dataset is the entailment one, that is the one for which the model produces the highest number of corrected predictions. More specifically, when a premise entails an hypothesis (461 total cases), 421 times

the model is able to recognize their relation, while 28 times (just the 0,06%) it predicts the relation of the pair as a neutral and just 12 times (0,04%) as a contradiction. Instead, for the less frequent class, that is the contradiction, the Bart model has still a quite good accuracy: 77,2% of times it correctly predicts the class (268 times in total), while 49 times (14%) it predicts entailment and 30 times neutral. Lastly, the class of the dataset for which the model achieved the lowest accuracy (61,7%) is the neutral one, even if it is not the less frequent class. It was slightly surprising that the most frequent kind of error of this model is when a pair of sentences labeled as neutral is predicted as an entailment (88 times, that means 22,4% over the total number of examples labeled as neutral).
Then, we computed the value of accuracy obtained on this experiment, along with the ones of precision, recall and F1score both with respect to the various labels and to the entire dataset. The overall accuracy on the whole dataset is equal to **72,1%**. The percentage values of the other metrics are reported in table 5.2.

	Precision	**Recall**	**F1score**
Contradiction	79%	66%	72%
Entailment	77%	90%	83%
Neutral	75%	70%	72%
Macro average	77%	75%	76%
Weighted average	77%	77%	77%

Table 5.2: Percentage values of the metrics about the Bart model's performance on the dataset.

As demonstrated also by the confusion matrix shown in figure 5.2, the label for which Bart is worst performing is the neutral. For the pairs of sentences originally labeled neutral, the precision (75%) is higher than the recall (70%). However, the overall accuracy of the model over our dataset is still not too low (**77,58%**) even if it achieves 90% of accuracy over the corpus on which it was trained (the MultiNLI). In order to have a more complete view of the predictions of the model, we should take into account the weighted average of the F1score (that is computed not a simple average but weighting the F1scores

of each class according to their frequencies on the dataset): its value is 77%.
Finally, we computed these metrics also with respect to the 12 linguistic categories defined in section 4.2. The figure 5.3 shows the respective accuracy values.

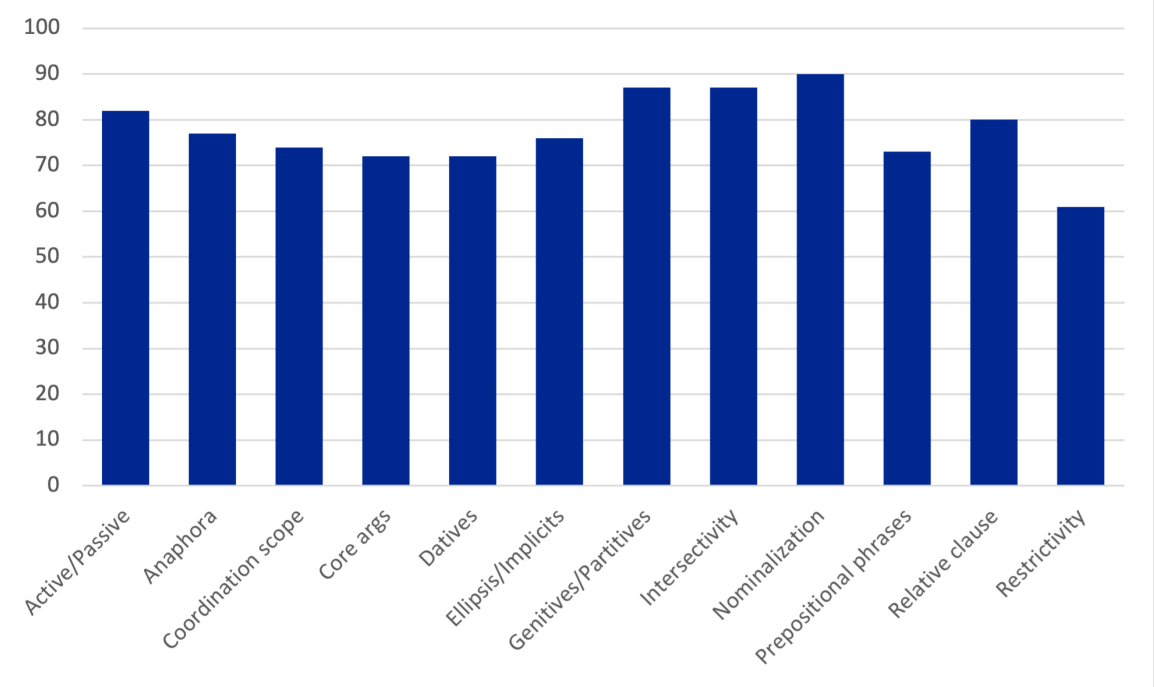

Figure 5.3: Percentage values of accuracy per category obtained by using Bart Large.

The linguistic category defined here for which the highest value of accuracy was obtained is the *Nominalization*: the accuracy is equal to 90%. Similar scores have been obtained for the classes of *Intersectivity* and *Genitives/Partitives* (87%), but also for *Active/Passive* and *Relative clauses* BART achieved a very high accuracy (higher than the average of the accuracy computed over the entire dataset). Instead, low performance values have been obtained over the *Datives* and *Core args* (72%) and *Restrictivity* (61%), which means that this model was not sufficiently able to recognize the entailment relation between pairs of sentences in which these kinds of phenomena are present.
In order to better analyze the performance of the model with respect to the linguistic categories, the figure 5.4 shows the error percentage of the model over those categories.

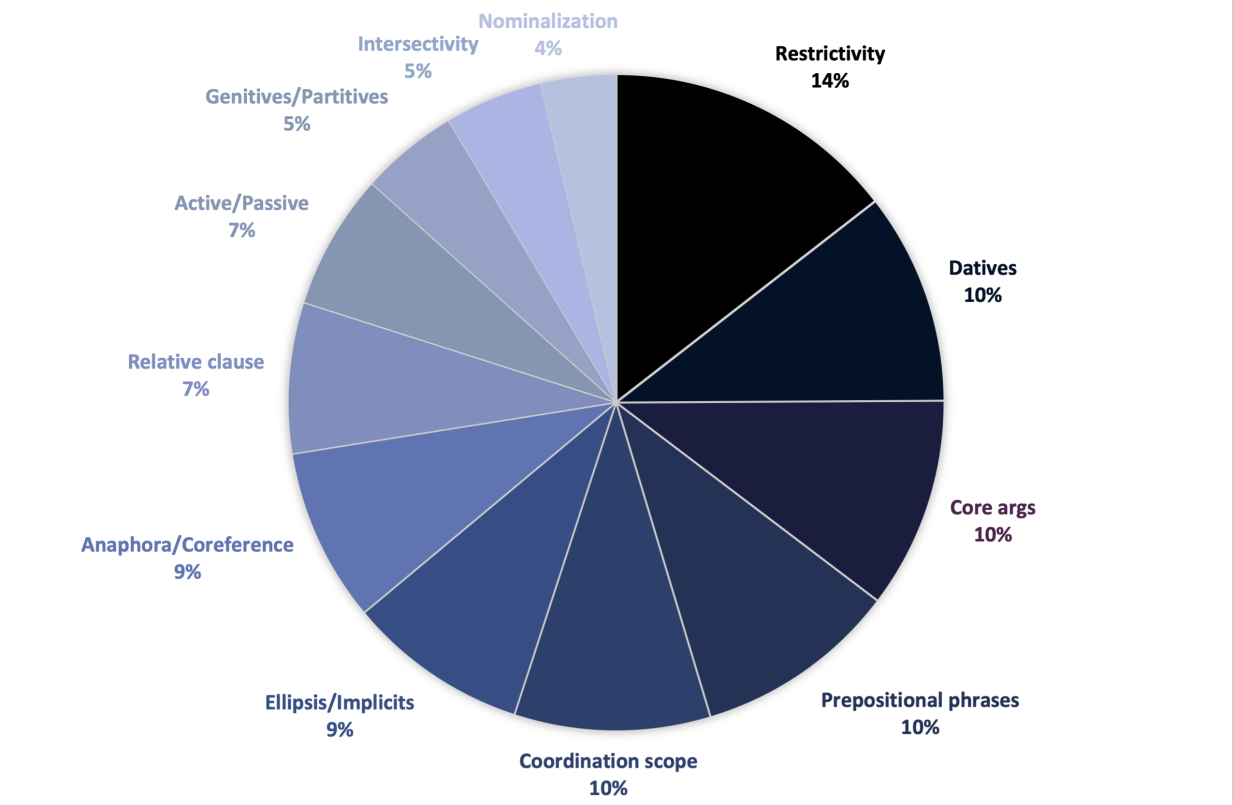

Figure 5.4: Percentage error values per category obtained using Bart Large.

5.4 RoBERTa

RoBERTa is a Transformer based model whose name stands for Robustly Optimized BERT Approach (Liu et al.; 2019). Since it can be challenging to determine which aspects of self-training methods like BERT, GPT and ELMo contribute the most on achieving their amazing results, Liu et al. studied the effects of the hyper-parameters tuning and training dataset size trough RoBERTa.

RoBERTa has a Transformer architecture with L layers. Each block uses N self-attention heads and has H hidden dimension. One of the difference with BERT is that RoBERTa has a longer training phase, during which a dynamic masking is performed. While in BERT a static masking is performed just once during the data processing, in RoBERTa the masking patterns are not the same for each training instance in every epoch but, every time the patterns are generated, they feed a sequence to the model. This is a crucial aspect when the

model is trained for more steps (with bigger batches) or with larger datasets. Moreover, the next sentence prediction objective was removed here. This choice comes from the fact that some researchers (Lample and Conneau; 2019, Yang et al.; 2019, Joshi et al.; 2019) already questioned the necessity of its loss: this was confirmed by results obtained by Liu et al. on the different trials of using RoBERTa with various alternative training formats. Furthermore, RoBERTa was trained with a batch size eight times larger for half as many optimization steps, thus seeing four times as many sequences in pretraining compared to BERT.

For our purposes, we used a version of RoBERTa trained by Facebook on the MultiNLI corpus, and made freely available on the HuggingFace webpage. This large version of the model has 355M parameters, 24 layers (with the hidden ones of size 1024) and 16 attention heads of size 64. We used such model through just a forward phase on our dataset. Once the model predicted the most probable label for each pair of sentences, we computed the resulting confusion matrix which is shown in figure 5.5.

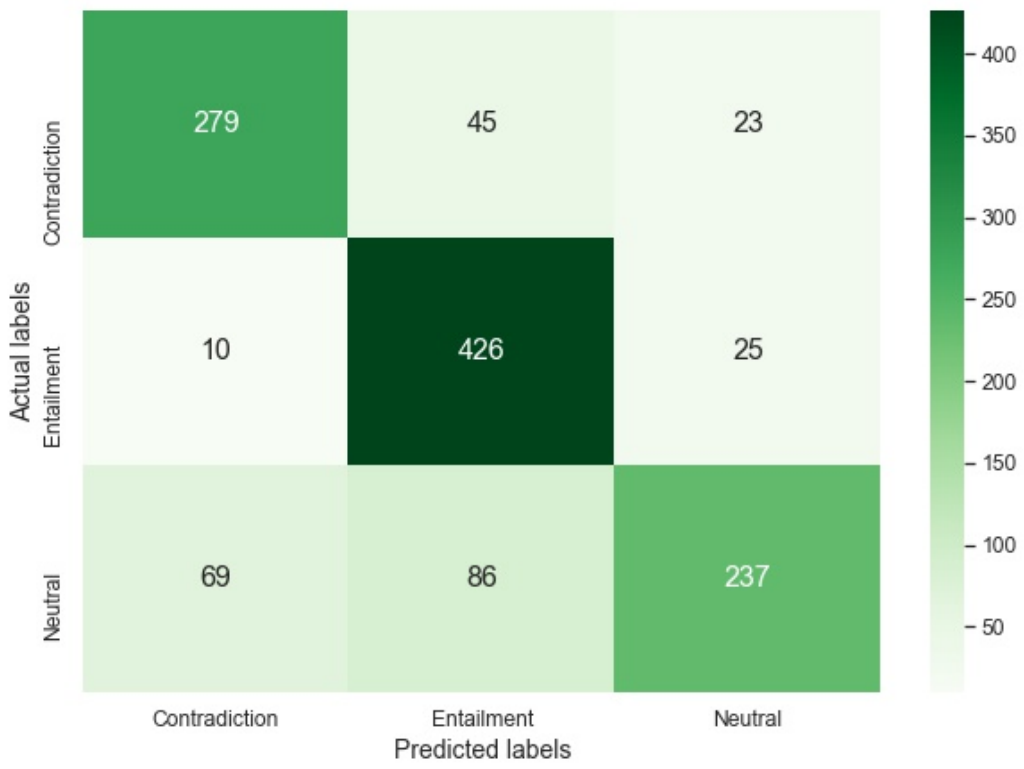

Figure 5.5: Confusion matrix for RoBERTa Large.

As we can see from the confusion matrix, also this model is very good at correctly predict the entailment relation between pair of sentences: the 92% of times (426 in total) the model goes right, while just 25 times it goes wrong classifying them as a contradiction or as neutral (10 times). Again, the class for which the model has greater difficulties is the neutral: 237 times it classify the pair correctly (60% over the total of neutral examples) but 86 times RoBERTa classify it as an entailment case (highlighting again the difficulties of these transformers towards this case) and 69 times as a contradiction. Instead, for the contradiction class, that is the less frequent one in our dataset, the model achieved still a quite good predictive performance: 279 times (over the 347 total cases) it classifies correctly, while when it goes wrong it tends to classify the pair as entailment more often (45) than it does with the class neutral (23).

Then, we computed the value of accuracy obtained on this experiment, along with the ones

of precision, recall and F1score both with respective to the single labels and the entire dataset. The overall accuracy on the whole dataset is equal to **78,5%**. The percentage values of the other metrics are reported in table 5.4.

	Precision	**Recall**	**F1score**
Contradiction	95%	72%	82%
Entailment	75%	95%	84%
Neutral	85%	73%	79%
Macro average	85%	80%	82%
Weighted average	84%	82%	82%

Table 5.3: Percentage values of the performance metrics of RoBERTa Large.

Then, we computed these metrics also with respect to the 12 linguistic categories defined in section 4.2. The figure 5.6 shows the percentage accuracy values over the 12 categories.

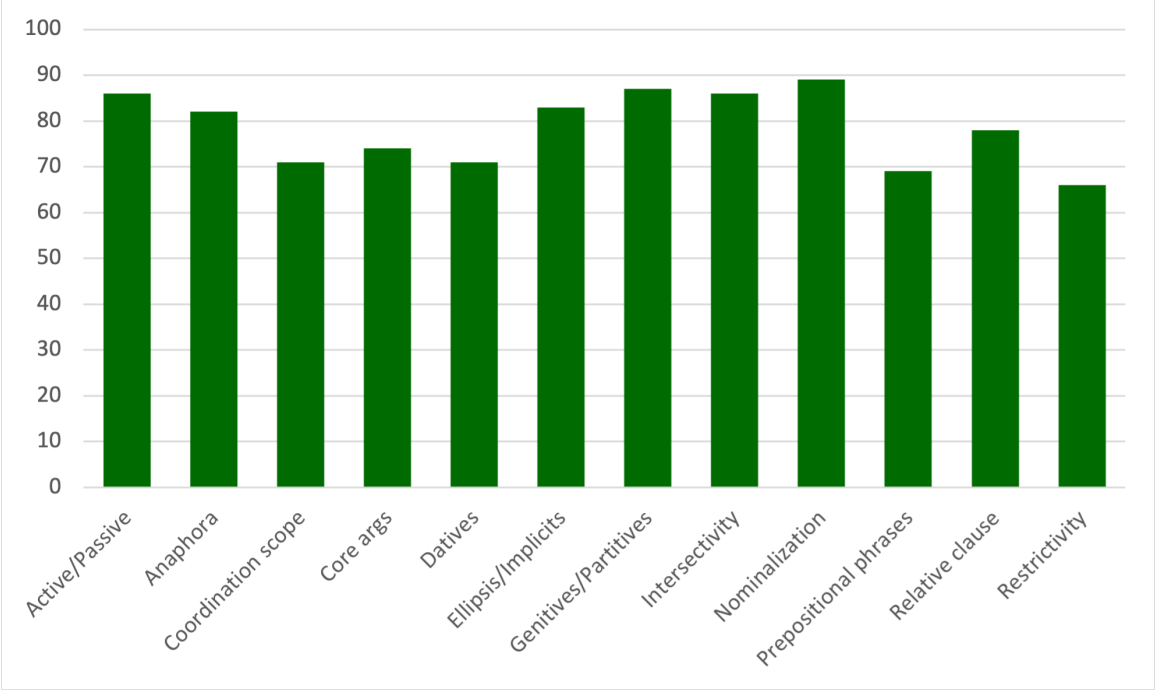

Figure 5.6: Percentage values of accuracy per category obtained by using RoBERTa Large.

RoBERTa achieved the higher value of accuracy on the category of *Nominalization*

(89%), followed by *Genitives/Partitives* (87%), *Intersectivity* and *Active/Passive* (86%). Instead, the lowest accuracy was obtained for the category of *Restrictivity*, for which the model was able to reach just an accuracy of 63%. Finally, we also computed the percentage error for each category while using RoBERTa Large on our dataset.

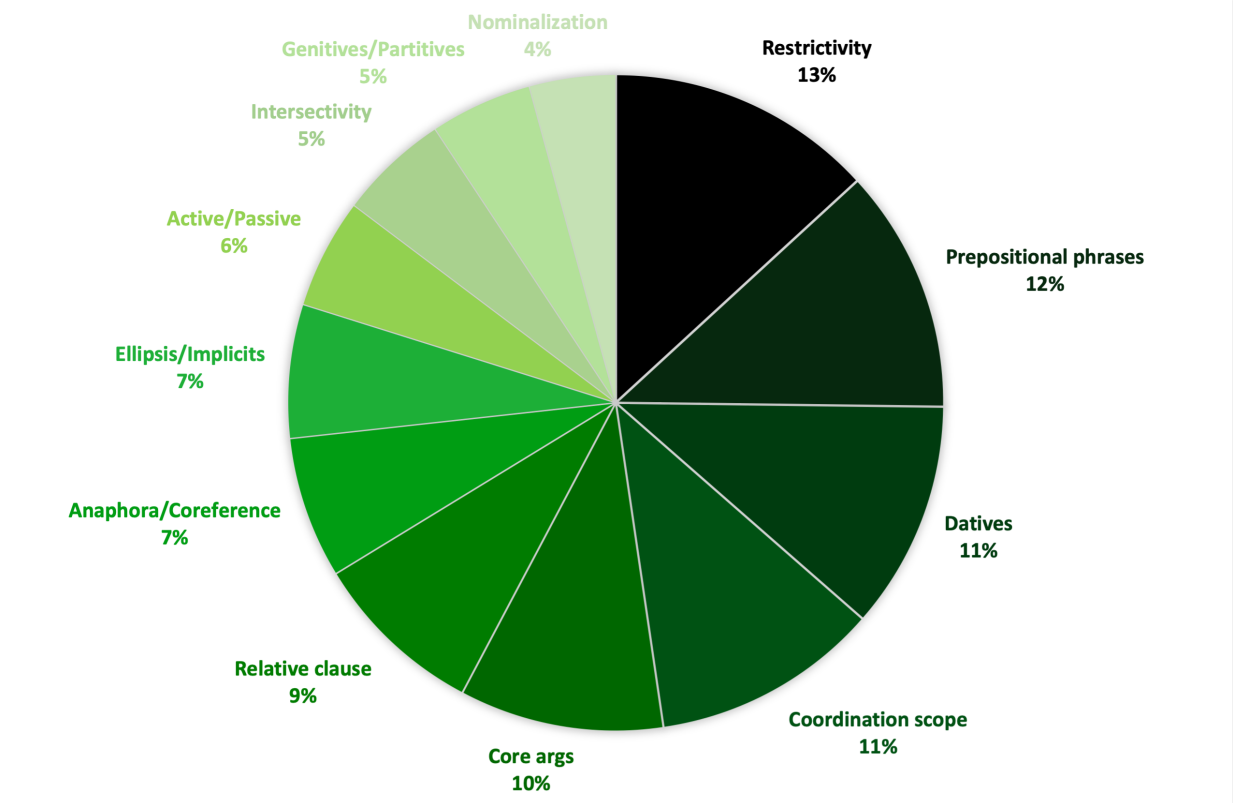

Figure 5.7: Percentage error values per category obtained using RoBERTa Large.

5.5 DeBERTa

DeBERTa stands for Decoding-enhanced BERT with disentangled Attention. It was introduced by Microsoft in 2021 (He et al.; 2021) [10]. DeBERTa improves the BERT and RoBERTa models using two novel techniques: the disentangled attention mechanism and an enhanced mask decoder. With the disentangled attention mechanism, each word is represented using two vectors that respectively encode its content and its position in the sentence, while with BERT each word is represented by just one vector. The attention weights among words are computed using disentangled matrices based on their contents

and relative positions respectively. This novel attention mechanism is motivated by the observation that the dependency between two words can change according to their position inside the sentence they occur.

The enhanced mask decoder is used to incorporate absolute positions in the encoding layer to predict the tokens that were masked in the model pre-training phase. Indeed, DeBERTa (unlike BERT) adds the content and position information of the context words for the masked language modeling task. The disentangled attention mechanism already incorporates contents and relative position for the context words, but it does not for their absolute positions which can be a crucial information for certain use cases. While BERT model incorporates absolute positions in the input layer, DeBERTa does it between the end of all the Transformer layers and the start of the *softmax* layer for the masked token prediction. By doing so, the absolute positions are used just as complementary information when decoding the masked words. The architecture of this model can be seen in figure 5.8.

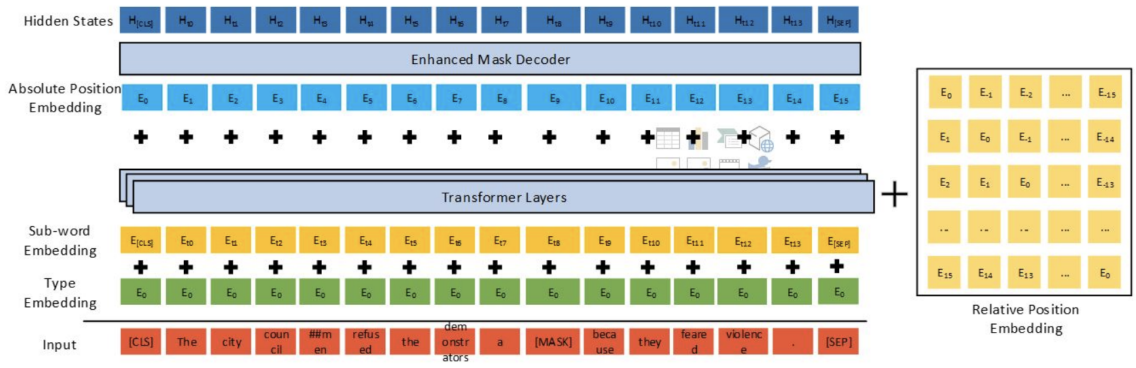

Figure 5.8: DeBERTa's architecture.

Moreover, He et al. proposed also a new virtual adversarial training method for fine-tuning the Pre-trained Language Models to downstream NLP tasks. This method has been demonstrated to be effective for improving the generalization capabilities of the models. Compared to RoBERTa-Large, a DeBERTa model trained on half of the training data performs consistently better on a wide range of NLP tasks, achieving a 91.1% of accuracy on the MultiNLI (+0.9%). DeBERTa surpasses human performance on SuperGLUE bench-

mark, achieving a 89.9% of macro-average score (against the 89.8%).

5.5.1 Version 1 Large

Firstly, we selected the first large version of DeBERTa trained by Microsoft on MultiNLI corpus. This version of the model is composed of 24 layers, 1024 units on the hidden layers, 16 attention heads of size 64 and it has a training time equals to 20 days. We used such configuration of the model for a forward phase over our dataset (as for the other experiments that we carried out). In figure 5.9 we reported the confusion matrix resulting from this experiment, with respect to our task of multi-classification (with labels being *entailment*, *neutral* and *contradiction*).

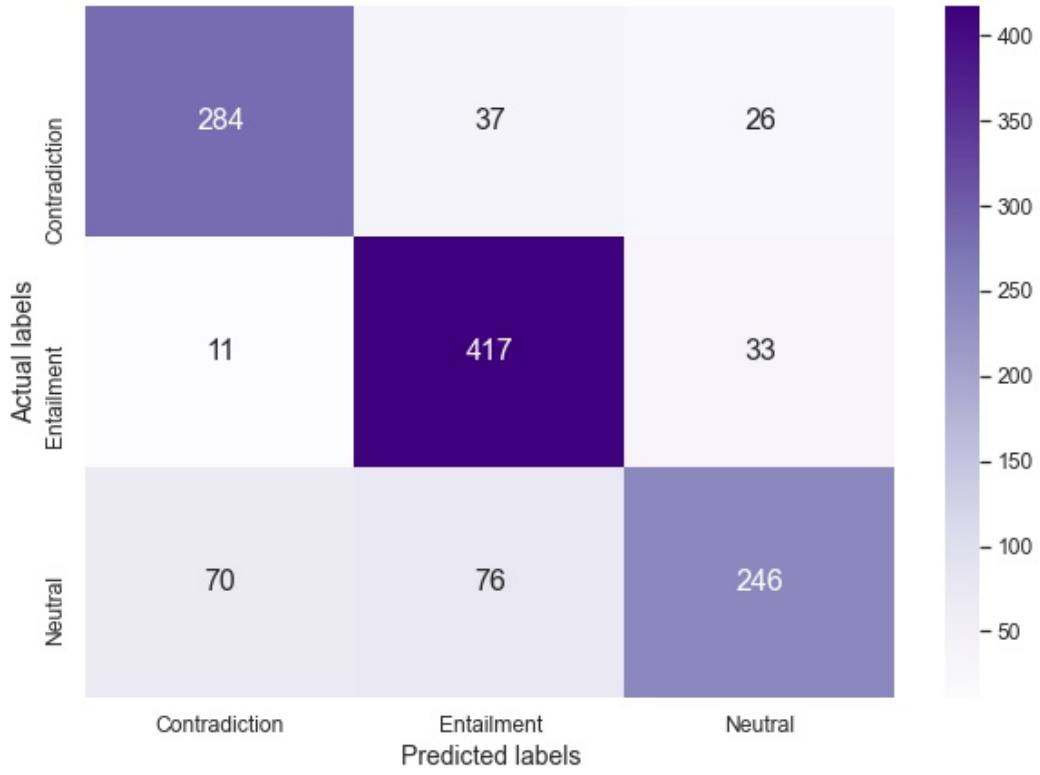

Figure 5.9: Confusion matrix for DeBERTa Large.

DeBERTa resulted to be particularly good at correctly predicting the entailment relation between the pairs of sentences, like also the previous models did. In particular, 417 times (over 461 actual cases of this class) the model made a correct prediction, while 33 times it wrongly classified the pair as neutral, and just 11 times as contradiction. So, for the entailment class the relative accuracy of the model is 90,5%. For the contradictory cases of the dataset, the model has still a quite good classification capability: out of 347 real cases of contradiction, 284 were correctly classified (the 81,8% of the total of this class). When the model was wrong in this category, it behave in a similar way giving either an output of entailment or neutral. So, the label for which the model has the worst performance is the neutral: 246 times it goes right (62,8%), but there are a great number of pairs actual labeled as neutral that are predicted as an entailment (76 in total) or as contradiction (70).
The overall accuracy of DeBERTa Large on the whole dataset is equal to **78,9%**. Then, we computed the accuracy, precision, recall and F1score both with respect to the various classes and to the entire dataset. The percentage values of these metrics are reported in table 5.4.

	Precision	**Recall**	**F1score**
Contradiction	78%	82%	80%
Entailment	79%	90%	84%
Neutral	81%	63%	71%
Macro average	79%	78%	78%
Weighted average	79%	79%	78%

Table 5.4: Percentage values of the metrics computed over the experiments using DeBERTa Large.

The predictive capabilities of this model are highlighted with the results reported in table 5.4: for the classes contradiction and entailment the recall is higher than the precision, while the opposite holds for the neutral class. In particular, the recall of the neutral class is very low compared to other ones, demonstrating that the model has some difficulties in

predicting it.

Then, in order to have a more complete view of the situation, we computed these metrics also with respect to the 12 linguistic categories defined in section 4.2. The figure 5.10 shows those accuracy values.

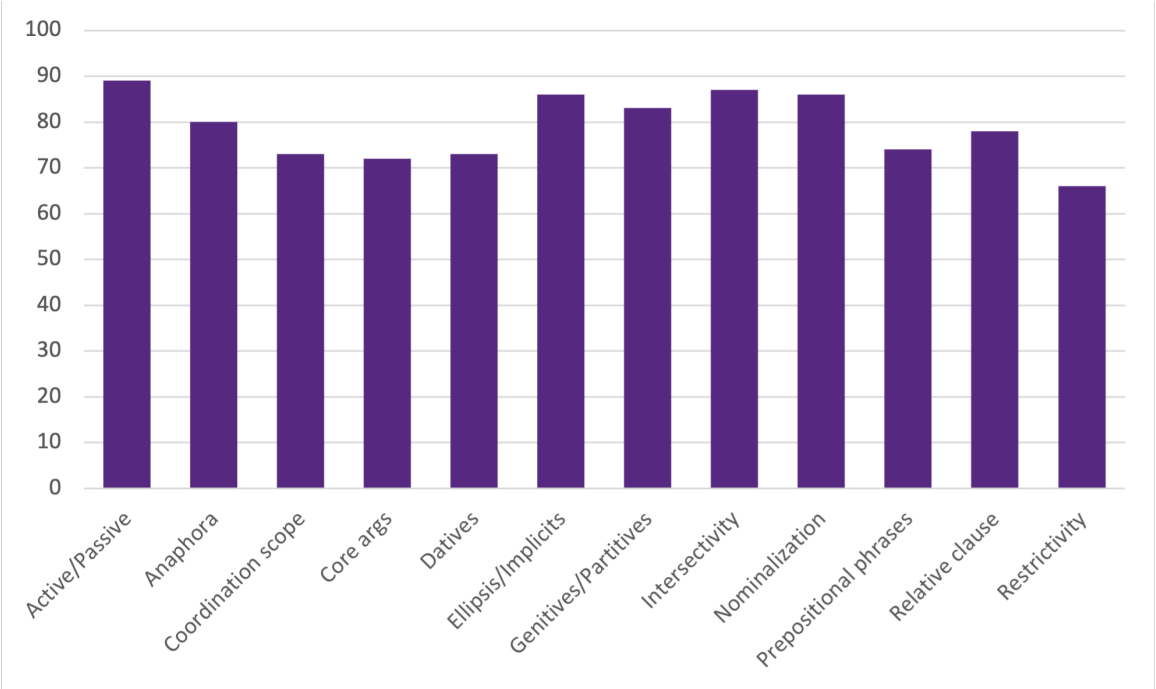

Figure 5.10: Percentage value of accuracy per category obtained by using DeBERTa large.

The highest value of accuracy (89%) was obtained over the pairs classified as *Active/-Passive*, demonstrating also in this case the Transformer is able is effective on the pair of sentences in which the entailment relation involve the active or passive form of the verb. Also for the categories of *Intersectivity* (87%), *Nominalization* and *Ellipsis/Implicits* an high accuracy was obtained (86% for the latter two). Instead, the lowest performance was obtained over the *Restrictivity* class (66%).

Finally, in order to better understand the predictive capabilities of this version of DeBERTa, we report in the figure 5.11 the percentages of the error obtained over all the 12 categories.

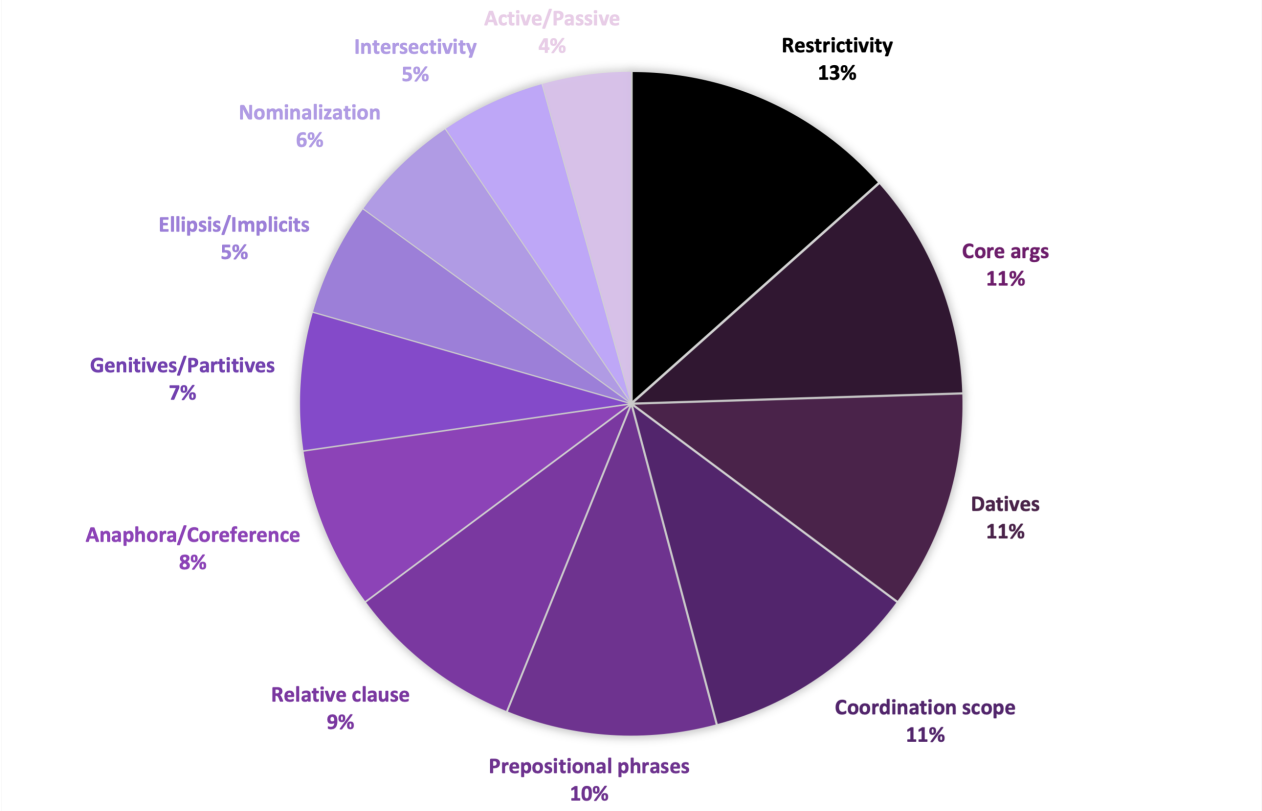

Figure 5.11: Percentage errors per category obtained by using DeBERTa Large.

5.5.2 Version 2 XLarge

The first version of DeBERTa trained on the MultiNLI corpus was released in 2020 by Microsoft. In 2021, He et al. released the code of DeBERTa v2, the 900M parameters and the model used for the SuperGlue submission of the research group.

In this version of DeBERTa different changes were made. First of all, the vocabulary was changed: the new vocabulary has size 128K and it was built from the training data. Instead of GPT2 tokenizer, here they use an unsupervised text tokenizer and detokenizer that implements subword units and unigram language model: such tokenizer was released by Google and called *sentencepiece tokenizer* [35]. Secondly, in the v2 a convolution layer was added to the architecture shown in figure 5.8: it allows to better learn the local dependency of input tokens. Furthermore, the position projection matrix is shared with the content projection matrix in the attention layer, since it was found that this can save parameters without affecting the model performance. These changes allow to scale the

model size to 900M and 1.5B, which significantly improves the performance of downstream tasks.

The v2 of DeBERTa for NLI tasks was released just with the XLarge and XXLarge versions. Here, we used the XLarge version, having 24 layers (the same as the version 1 Large of DeBERTa whose results on our dataset were described in the subsection 5.5.1) with the hidden ones of size 1536 (against of the 1024 of the other version used here). Also for this model, we performed a forward phase. Figure 5.12 shows the confusion matrix resulting from the classification output of the model.

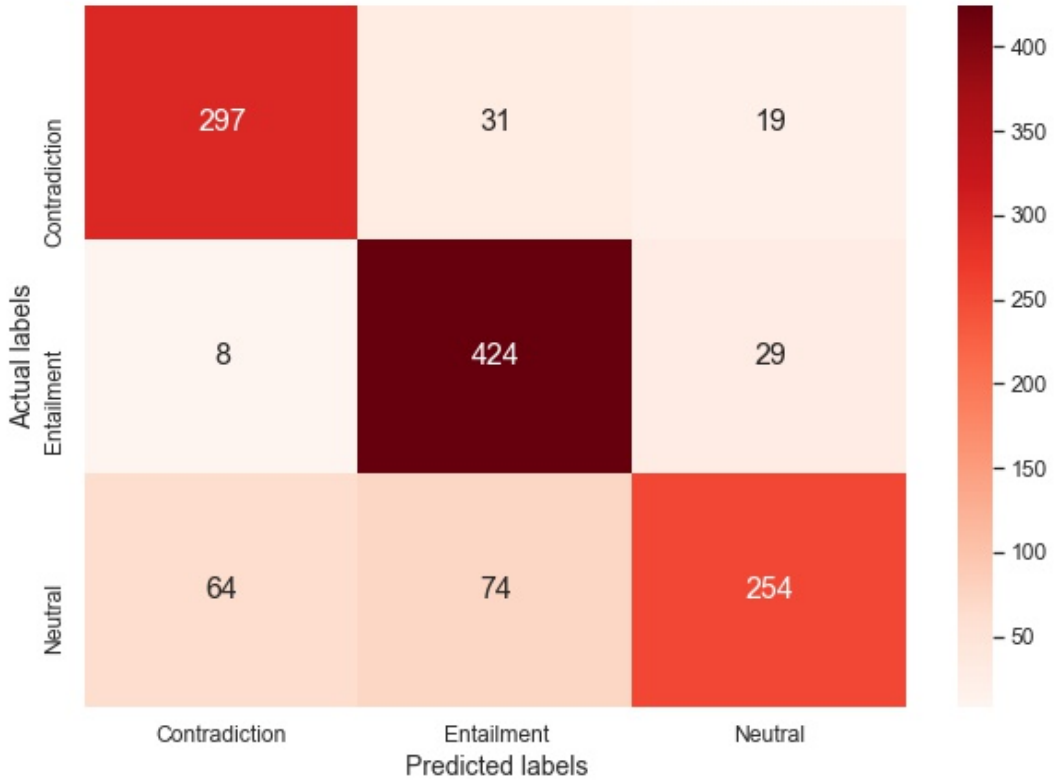

Figure 5.12: Confusion matrix for DeBERTa V2 XLarge.

This version of DeBERTa resulted to be very good at predicting the entailment relation between premises and hypotheses: 424 times this class is predicted correctly (corresponding to the 91,9% of the class), while when it goes wrong for this class the greater of times

it classifies the pair as a neutral case (29 times). For the class contradiction, it has also a quite good classification capabilities (85,6% of times it goes right, that means for a total of 297 times). Instead, the class neutral is the most difficult to be correctly predicted for this model. In particular, the model tends to predict those items as an entailment for 18,9% of times (74 times in our dataset), while for 16,3% of times it predicts them as a contradiction (64 times in total).

Given these data, the overall accuracy obtained by using this model on our dataset is **81,25%**. In order to better analyze the classification capabilities of v2 of DeBERTa XLarge, we show in table 5.5 the resulting values of precision, recall and f1score computed with respect to the classes of the dataset as well as their macro and weighted average.

	Precision	**Recall**	**F1score**
Contradiction	80%	86%	83%
Entailment	80%	92%	86%
Neutral	84%	65%	73%
Macro average	82%	81%	81%
Weighted average	82%	81%	81%

Table 5.5: Percentage values of the metrics about the Bart model's performance on the dataset.

Finally, we computed these metrics also with respect to the 12 linguistic categories defined in section 4.2. In particular, the figure 5.13 shows the accuracy values for each of the 12 categories of linguistic phenomena.

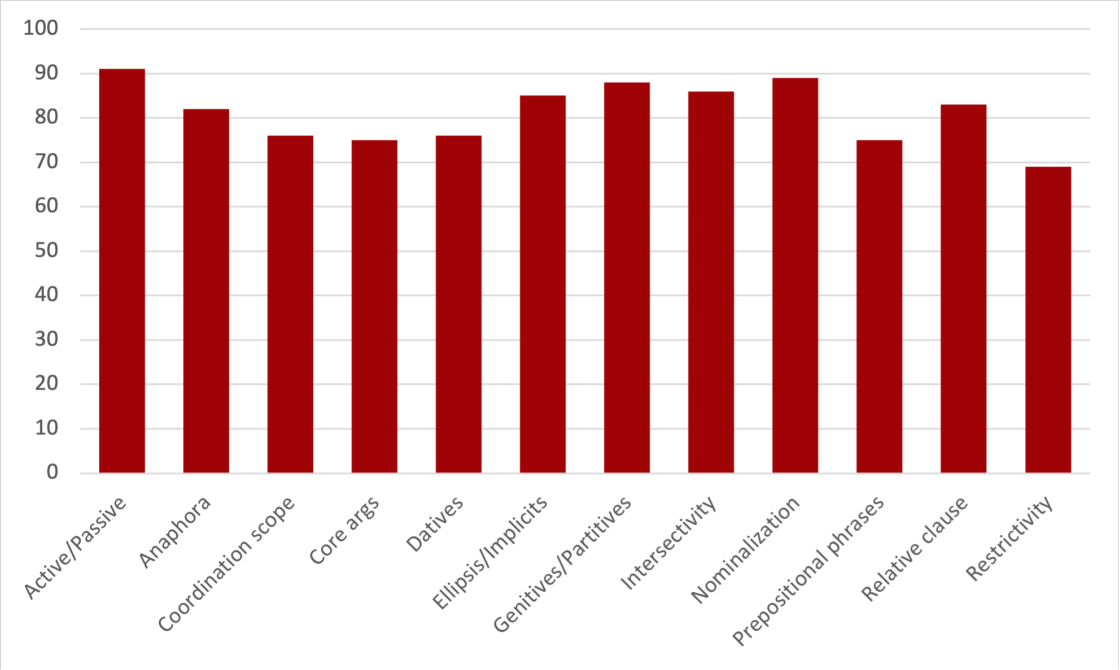

Figure 5.13: Percentage values of accuracy per category obtained by using the version 2 of DeBERTa XLarge.

For some of the 12 categories we obtained a very high value of accuracy. Firstly, for *Active/Passive* DeBERTa achieved an accuracy equals to 91%. This value is followed by: *Nominalization* (89%), *Genitives/Partitives* (88%), *Intersectivity* (86%) and *Ellipsis/Implicits* (85%). For the remaining categories, the accuracy is still quite good. Instead, the category for which the model has encountered more difficulties is *Restrictivity*: the accuracy obtained here is 69%.

Finally, we computed also the percentage errors obtained by the model along each category: those values are shown in figure 5.14.

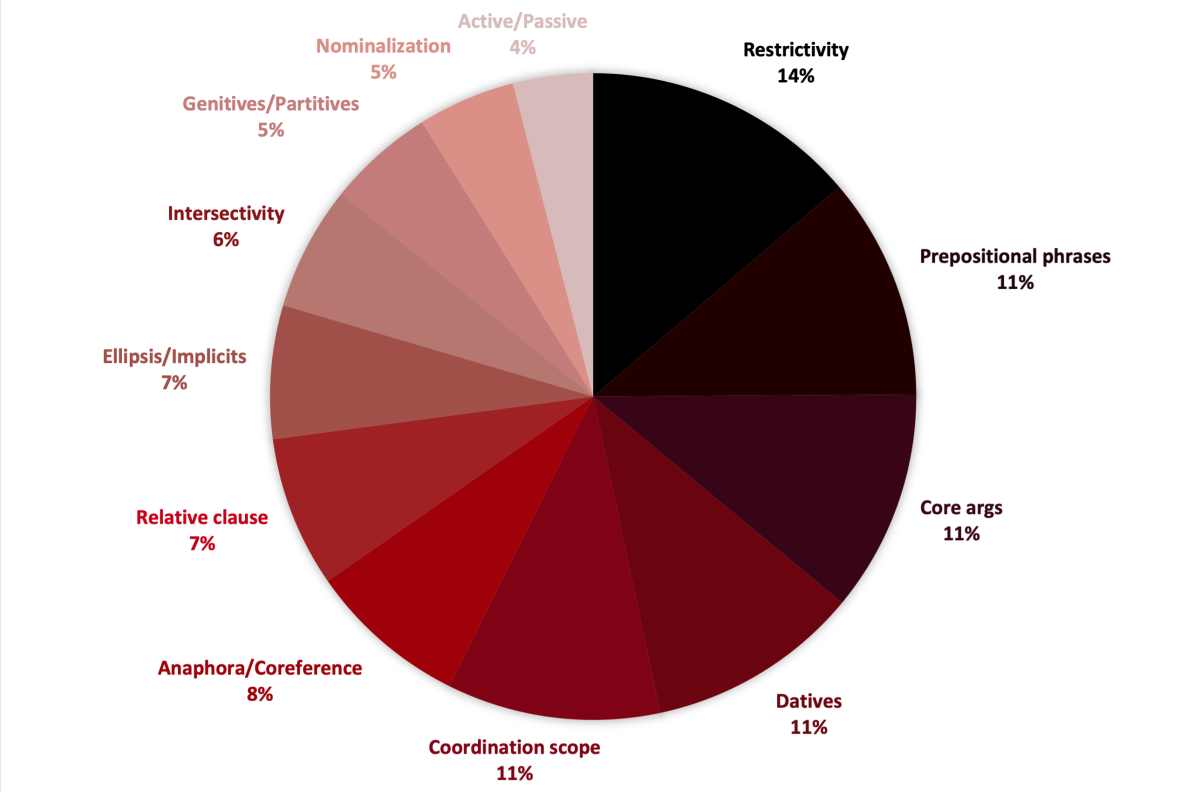

Figure 5.14: Percentage errors per category obtained by using the version 2 of DeBERTa XLarge.

5.6 Results comparison

Once we used those Transformer models as described in section 5.2 and computed their evaluation metrics over our dataset, we compared their results. In table 5.6 we report the precision, recall and f1score for all the 4 models, computed as a weighted average of their values over the classes of the dataset.

Model	Precision	Recall	F1score
Bart Large	77%	77%	77%
RoBERTa Large	84%	82%	82%
DeBERTa V1 Large	79%	79%	78%
DeBERTa V2 XLarge	82%	81%	81%

Table 5.6: Precision, recall and f1score of the models computed over our dataset.

Looking at those metrics, the model best performing on our dataset seems to be RoBERTa Large, which achieves very high precision and recall, and an f1score equals to 82%. Its results are not far away from the results obtained by DeBERTa V2 XLarge (81%). However, the version of the latter model has a bigger architecture than the version of RoBERTa. This choice of using a model with a bigger number of layers (with more nodes) and of attention heads was done since the second version of DeBERTa trained on the MultiNLI has been released by Microsoft just in the XLarge and XXLarge versions. On the other side, this allows us also to measure how much the size of the model affects the final results on a dataset that is not huge: the difference between a large pre-trained model and an even larger one seems to be not so big, although considering that are also other differences between them than their sizes.

As long as the models we chose are all pre-trained on the same corpus, that is theMultiNLI, we are able also to compare the results we obtained on our dataset with the results the researcher obtained on the MultiNLI. The table 5.7 reports the accuracy obtained by each of the selected model on our dataset and on the corpus on which they were trained on.

Model	Accuracy Dataset	Accuracy MultiNLI
Bart Large	77,58%	89,9%
RoBERTa Large	78,5%	90,2%
DeBERTa V1 Large	78,9%	91,1%
DeBERTa V2 XLarge	**81,25%**	**91,6%**

Table 5.7: Accuracy obtained by the models over our dataset and over the MultiNLI.

As it could be predicted, all the models tried on our dataset achieved an accuracy lower than the accuracy that was obtained on the training corpus. However, those accuracy values are still quite satisfying even if they are much lower: the differences between the accuracy on the MultiNLI and on our dataset are between 12,3 and 10,35. We have to consider that in the dataset may occur some phenomena that were not present in the training set or not sufficiently represented in it: this is a well-know problem in Natural Language Processing and Machine Learning, but that can be masked by larger and larger datasets. Moreover, our dataset take into account different phenomena that can be difficult to be understand also by human readers since they can cause some ambiguities.
This hypothesis was somehow considered also in light of the different performances of the models over the 3 classes of the dataset and the 12 categories of the linguistic phenomena considered for this work. In figure 5.15 we show the accuracy obtained by using each model on our dataset, with respect to the 3 classes. For all of the Transformers used in our experiments, the lowest accuracy was obtained with respect to the Neutral class even if it is not the less frequent one. So anyway, the overall performances were significantly affected by the models' difficulties of predicting this class.

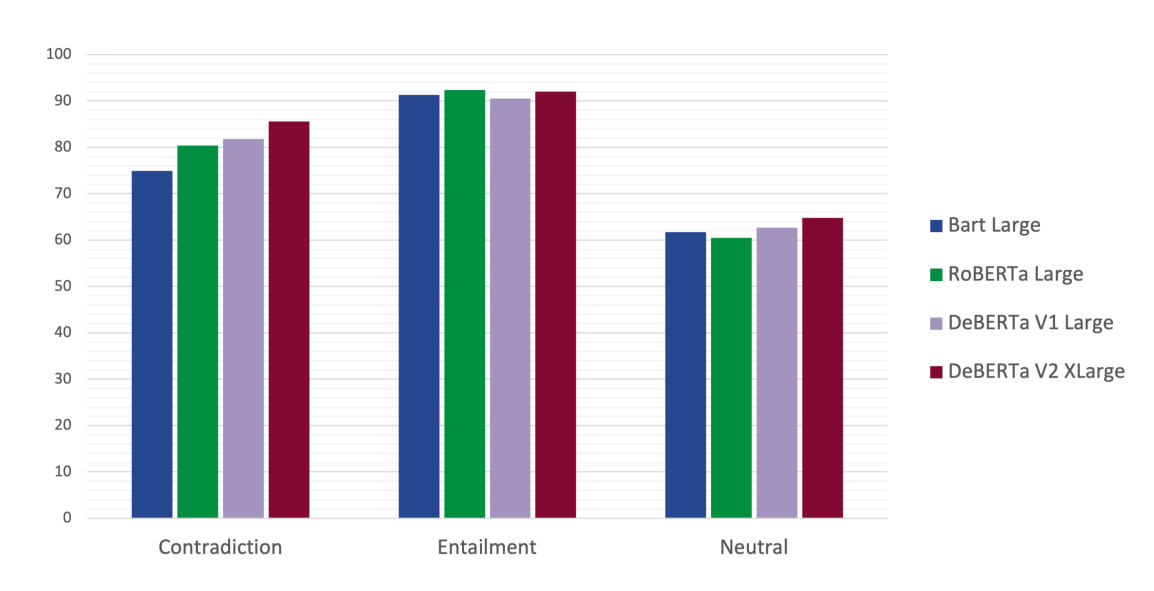

Figure 5.15: Accuracy of the models with respect to the classes of the dataset.

Figure 5.16 shows the accuracy of each model computed with respect to each fine-grained category. Also in this case there is a good variance of the results along the categories of the dataset, even though they are perfectly balanced (for each category there are exactly 100 pairs). This highlights that, regardless of the specific model considered, its size and its architecture, those Transformers are more capable to recognize a specific type of entailment relation than others. More in details, the fine-grained for which we obtained a sufficiently high accuracy in all the cases are: *Nominalization*, *Intersectivity*, *Genitives/-Partitives* and *Active/Passive*. In these categories the accuracy was always higher than 80%. A good results in almost all cases were obtained for *Relative clauses*, *Anaphora* and *Ellipsis/Implicits*. On the other hand, the models were not sufficiently able to correct analyze the pairs of sentences belonging to *Restrictivity* (for which the accuracy was never higher than 70%), *Coordination scope*, *Core args* and *Datives*.

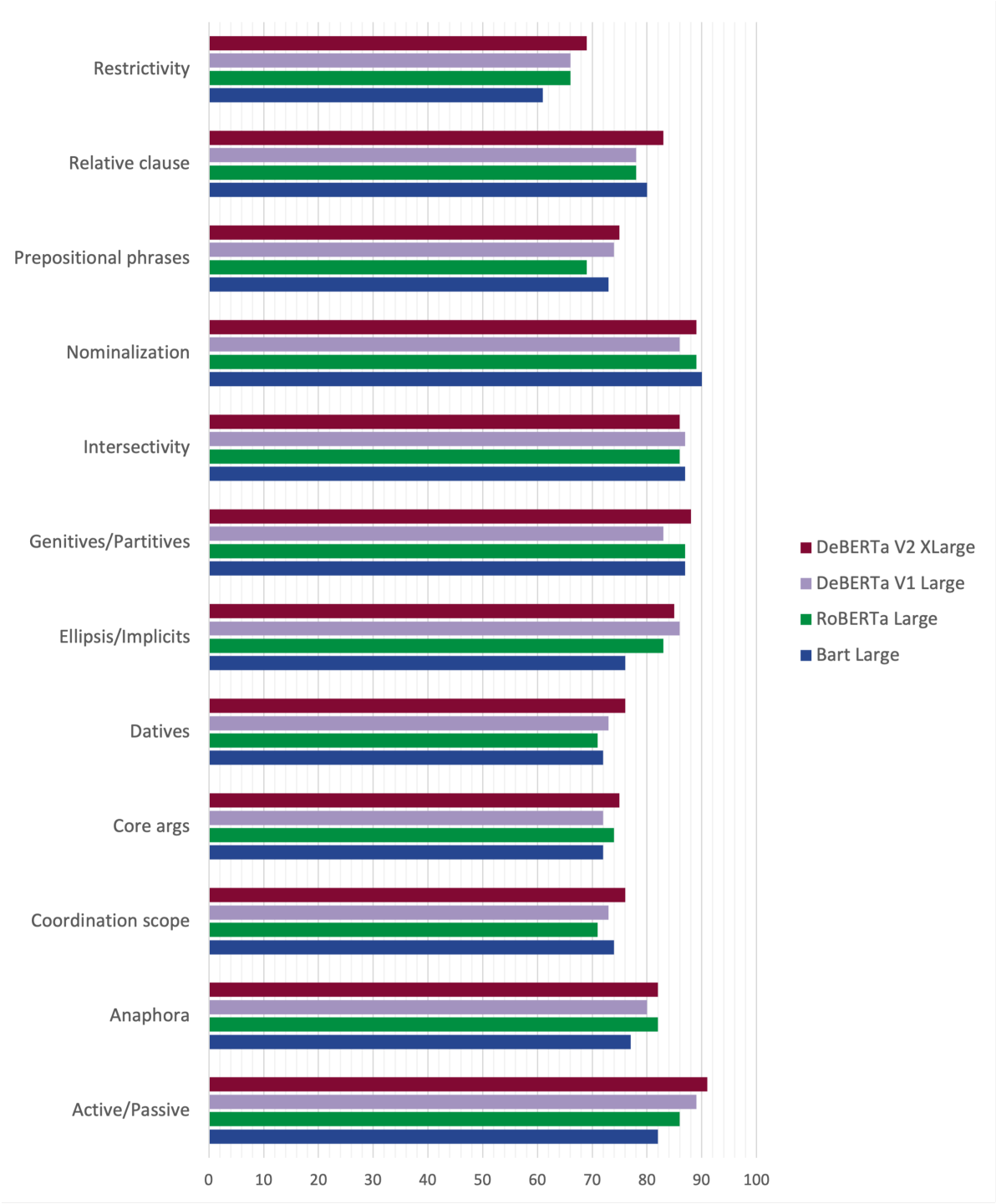

Figure 5.16: Accuracy of the models with respect to the fine-grained categories.

From this comparison of the accuracy values, it results that the worst performing model is Bart Large. Instead, RoBERTa Large and DeBERTa V1 Large achieved results that are quite similar in almost aspects. The model that achieved the highest accuracy is DeBERTa

V2 XLarge maybe thanks to the adjustments with respect to the previous version of the same model or maybe thanks to the fact that it is a larger model than the others (or also maybe thanks to the combination of these factors). However, since our dataset is not perfectly balanced and in order to have a more complete view of the differences on the results of the 4 models than the one offered by the accuracy, we take into account also their precision, recall and f1scores.

5.7 Errors analysis: DeBERTa

Comparing the results of the selected models, we have seen that the one performing better on our dataset is the second version of DeBERTa in its XLarge form. In order to better analyze its generalization capabilities we report for each fine-grained category some examples that were wrongly classified by the model.

Active/Passive

For the fine-grained category of *Active/Passive* DeBERTa V2 XLarge achieved a very high accuracy (91%). In the few errors produced, were not found any specific patterns of errors. The errors produced are linked to the pairs of sentences that could cause more ambiguities. In table 5.8 we report two examples, both belong to the label neutral but classified as entailment, which is the most common kind of error.

Premise	**Hypothesis**	**Actual class**	**Predicted class**
Only 32% of Russians trust their president, according to a recent poll.	Only 32% of Russian are trusted by their president, according to a recent poll.	Neutral	Entailment
Twitter has temporarily suspended Republican lawmaker Marjorie Taylor Greene for posting ”misleading” information about coronavirus.	Misleading information about coronavirus have been suspended.	Neutral	Entailment

Table 5.8: Examples of a misclassified pair of sentences belonging to the *Active/Passive* category.

Anaphora/Coreference

The accuracy obtained by DeBERTa V2 XLarge for the fine-grained category of *Anaphora/-Coreference* is 82%. The simplest cases (for example when the referent is quite clear or near to the second item referring to it) were mostly correctly classified. Instead, the model encountered some problems while predicting the label of those pair of sentences where one referent could be misunderstood also by some native speakers. In table 5.9 we report some examples with the actual label and the predicted one.

In the first case reported on the table, the word *it* of the premise refers to the action *told her not to go on vacation* since also the subject of the verb *regretted* is *he* and so refers to *Marc*. So, the pair should be classified as a neutral because in the premise we do not have any information about the fact that Marc gone on vacation and Marc regretted to had gone on vacation. However, the model is not able to capture this information and classified this example as entailment.

In the second example of the table 5.9, the actual class is entailment while the model predicted neutral. In the premise text, the expressions *harassment*, *trolling* and *state surveillance* are all intended as *online threats, undermining journalists' work*. In the hypothesis,

Premise	**Hypothesis**	**Actual class**	**Predicted class**
Marc told her not to go to on vacation but then he regretted it.	Marc regretted to not have gone on vacation.	Neutral	Entailment
The organisation also lists online threats, such as harassment, trolling and state surveillance, as undermining journalists' work across the continent.	The organisation also lists harassment, trolling and state surveillance, as undermining journalists' work across the continent, even in countries where freedom is held in high regard.	Entailment	Neutral

Table 5.9: Examples of a misclassified pair of sentences belonging to the *Anaphora/Coreference* category.

those expressions still refers to the cause *undermining journalists' work* even if are not specified as *online threats*. The error of the model is maybe due to the fact that another information is added (*even in countries where freedom is held in high regard*): however, this should not change the relation between the two sentences since it is an extra information. This latter example of error produced in this category underlies that the misclassified pairs are the more complex ones, that could be discussed also by humans since there is a quite fine line on its classification.

Coordination scope

For the category of *Coordination scope* the model produced a bigger number of errors: 76 examples over 100 were correctly classified. The errors produced here are mainly due to the substitution of the coordinating conjunction of the premise. For example, the first row of table 5.10 highlights one of these cases.

From the premise of the first pair of example we grasp that on Saturday, Tom and Alice

will both sing and dance, while on the hypothesis it said that they will sing or dance, so they will do just one of the two actions: this is a clear case of contradiction, but the model does not grasp it and instead classifies the example as an entailment.
The other example in table 5.10 is a bit more complex than the first one because it is more complex the coordination between the conjuncts and the main verb they both refers to. In the second example, the premise told us that the reason why Lewis has to stay home is to look after his little brother. Instead, the fact that Lewis is sick is not related to the fact that he has to stay home. So, the hypothesis is in a contradictory relation with the premise because it tells that the fact that Lewis is not sick is related to the fact he has to stay home.

Premise	**Hypothesis**	**Actual class**	**Predicted class**
Tom and Alice will sing and dance on Saturday.	Tom and Alice will sing or dance on Saturday.	Contradiction	Entailment
Lewis has to stay home not because he is sick, but rather to look after his little brother.	Lewis has to stay home because he is not sick.	Contradiction	Entailment

Table 5.10: Examples of a misclassified pair of sentences belonging to the *Anaphora/-Coreference* category.

Core args

The accuracy obtained by the model on the *Core args* category is 75%, so it is not the highest one. In this category the models made different kinds of errors: some of them are reported in table 5.11.

Premise	**Hypothesis**	**Actual class**	**Predicted class**
Luke saw the the dog.	The dog saw Luke.	Neutral	Contradiction
Jake is Charles's brother.	Charles is Jake's brother.	Entailment	Contradiction
She left the house just after us.	She wasn't in the house.	Contradiction	Entailment

Table 5.11: Examples of a misclassified pair of sentences belonging to the *Core args* category.

In the example on the first row, the arguments of the verb of the premise sentence are inverted. In this specific case, this cause an hypothesis that is not related to the first one. The fact that Luke saw the dog, does not give us any information whether also the dog saw Luke. However, the model predicts this example as a contradiction.
Another type of error of DeBERTa in this category is due to the fact that the model might don't have some information about the world. In order to correctly classify the pair of sentences in the second row of table 5.11, the model should know that if a person A is the brother of a person B, than B is the brother of A. Other kind of relations (such as for the relation of parenthood) are not symmetric like this one. However, the model's error here may be also due to the presence of the genitive case.
From the premise of the last example on the table, we can understand that we left the house and this happened before she left the house. So, the information on the hypothesis (that she was not in the house) contradicts the premise, since she must have been in the house in order to left it. However, the model predict this as an entailment.

Datives

The fine-grained categories of *Datives* could be not so difficult to be analyzed since the dative case usually appears in English in a standard grammatical construction. However, the accuracy obtained by the model on this category is just 76%. In table 5.12 we report some examples of this category that were misclassified: more precisely, they both were

labeled neutral but one is predicted as entailment, contradiction the other.

Premise	**Hypothesis**	**Actual class**	**Predicted class**
Mr. Johnson lend her money.	Mr. Johnson lend you money.	Neutral	Entailment
Your mum should tell you the truth.	You should tell your mum the truth.	Neutral	Contradiction

Table 5.12: Examples of a misclassified pair of sentences belonging to the *Datives* category.

In the first case the expression *her* in the dative case was substituted with *you*: this is neutral because the fact the Mr. Johnson lend money to her is not related to the fact the Mr. Johnson lend money to you. Instead, in the second case of error, the subject and the dative were inverted in the hypothesis: also in this case the actual label is neutral, however the model predicts as the premise contradicts the hypothesis.

Ellipsis/Implicits

The model achieved an accuracy of 85% for the category of *Ellipsis/Implicits*, so one of the highest. In table 5.13 we report three pairs of sentences belonging to this fine-grained category that were misclassified. In the first example, in the hypothesis an extra information (the verb *likes*) is added where the verb was implicit. This extra information put the hypothesis in a neutral relation with the premise (from the premise we know that Marc's sister can play the violin but we don't have any information about whether she likes it or not). The second example is predicted as a contradiction rather than neutral: from the premise we entail that Biden and Putin talked about a possible summit last week and that the Kremlin said it would take time to organise this possible summit, we don't know if the Kremlin will actually organized it. Finally, the third case is probably one of the most difficult since the presence on the hypothesis of the negation of a part of the premise.

Premise	**Hypothesis**	**Actual class**	**Predicted class**
Marc can play the piano, his sister the violin.	Marc can play the piano, his sister likes the violin.	Neutral	Entailment
Biden and Putin discussed a possible summit last week, but the Kremlin has said it would take time to organise.	The Kremlin has said it would organise the summit.	Neutral	Contradiction
They achieved more in the past 10 days than in the whole 2020.	They did not achieved more in the whole 2020.	Entailment	Contradiction

Table 5.13: Examples of a misclassified pair of sentences belonging to the *Ellipsis/Implicits* category.

Genitives/Partitives

For the category of *Genitives/Partitives* DeBERTa achieved a very high accuracy (88%). The few errors that are made here are due to more complex sentences such as the one reported in table 5.14 that was predicted as entailment rather than a contradiction.

Premise	**Hypothesis**	**Actual class**	**Predicted class**
President Vladimir Putin's spokesman Dmitry Peskov said moving troops across Russian territory was an "internal affair".	The spokesman of Dmitry Peskov said moving troops across Russian territory was an "internal affair".	Contradiction	Entailment

Table 5.14: Examples of a misclassified pair of sentences belonging to the *Genitives/Partitives* category.

Intersectivity

The accuracy of the model with respect to the category of *Intersectivity* is equal to 86%. The model produced just few errors for some difficult cases, such as the one reported in table 5.15. This example should be a contradiction since families and players both ran for cover, some of them hiding in the dugouts and some others rushing for the exits or nearby buildings, so saying that families just run for hiding in the dugouts while players just rushed for exits is a contradiction.

Premise	**Hypothesis**	**Actual class**	**Predicted class**
Families and players alike ran for cover, some hiding in the dugouts, some rushing for the exits or nearby buildings.	Families ran for cover, hiding in the dugouts, players rushed for the exits or nearby buildings.	Contradiction	Entailment

Table 5.15: Examples of a misclassified pair of sentences belonging to the *Intersectivity* category.

Nominalization

Also for the *Nominalization* category a very high accuracy was achieved by DeBERTa V2 XLarge (89%). As reported in section 4.2.3, this phenomenon usually occurs in some standard ways since these nouns usually have some specific suffixes. An example of pair of sentences belonging to this category that was wrongly classified is reported in table 5.16

Premise	**Hypothesis**	**Actual class**	**Predicted class**
Laura apologized for not having told him the truth.	He received Laura's apologies.	Entailment	Neutral

Table 5.16: Examples of a misclassified pair of sentences belonging to the *Nominalization* category.

Prepositional phrases

The accuracy obtained for the *Prepositional phrases* is equal to 75%. In table 5.17 we report two pairs with the respective actual label and the predicted one. In the first example reported the actual label is neutral since we do not know whether Alice is John's sister nor we talk about the school in the premise sentence: maybe this lack of information lead the model to predict the pair as a contradiction. Instead, the second pair was classified as entailment rather than neutral. Also in this case, in the premise we have a lack of information that is present on the hypothesis: we do not know if the person who is speaking is behind the scenes in that moment, so we do not have enough information to classify the example as an entailment (and the same holds for the label contradiction).

Premise	**Hypothesis**	**Actual class**	**Predicted class**
John goes to the park with Alice.	John goes to the school with his sister.	Neutral	Contradiction
They will be behind the scenes during the show.	They will be there during the show.	Neutral	Entailment

Table 5.17: Examples of a misclassified pair of sentences belonging to the *Prepositional phrases* category.

Relative clauses

The accuracy obtained by DeBERTa V2 XLarge for the fine-grained category of *Relative clauses* is 83%. Some of the errors produced in this category by the model regard pairs of sentences in which the pronoun *whose* is present on at least one sentence. An example of this case is reported in table 5.18. In this specific example, the pronoun *whose* in the premise sentence refers to the expression *data scientist* and not to the expression *campaign*. For this reason, the pair should be labeled as neutral rather than entailment (we do not specify what the job of the campaign is on the premise text so we can not infer nor contradict this information).

Premise	**Hypothesis**	**Actual class**	**Predicted class**
The campaign was uncovered in August 2018 by a data scientist, whose job involved combatting fake engagement.	The job of the campaign was combatting fake engagement.	Neutral	Entailment

Table 5.18: Examples of a misclassified pair of sentences belonging to the *Relative clauses* category.

Restrictivity

The lowest accuracy per category was obtained by DeBERTa on the one of restrictivity: for this category the accuracy is equal to 69%. Understand whether a modifier has a restrictive use or not can sometimes be difficult also for native speakers since the same modifier can be used in different ways in different contexts and there is not a formal rule to distinguish between a restrictive and a non restrictive usage. In table 5.19 were reported two misclassified examples belonging to this category with the actual label and the one predicted by the model.

Premise	**Hypothesis**	**Actual class**	**Predicted class**
Karl and Lewis are travel lovers.	Karl and Lewis are lovers.	Neutral	Entailment
I don't want to hangout with people who don't show respect for the world environment.	I don't want to hangout with people.	Neutral	Entailment

Table 5.19: Examples of a misclassified pair of sentences belonging to the *Restrictivity* category.

6. Conclusions

In this thesis we presented the work whose main aim was to construct an English dataset for addressing a Natural Language Inference task. The Natural Language Inference is a challenging subtask of the Natural Language Understanding that is crucial for and related to other key tasks dealing with languages. The goal of this kind of task is to recognize the entailment relation between pairs of sentences, composed of a premise and an hypothesis. In the construction of the dataset we put our focus on a set of linguistic phenomena that are strongly related to the predicate-argument structure of sentences. In order to identify such phenomena we performed an analysis of the state-of-the-art dataset for Natural Language Inference and we decided to use a structure similar to the one of the Glue benchmark. We built sentences that were as simple as possible, with just one phenomenon involved in their entailment relation. This choice allow us to perform some experiments to analyze what kind of information some state-of-the-art models are able to grasp.

The experiments consisted of a single feed-forward phase performed using some Transformer based models that were pre-trained on the MultiNLI corpus and freely available on the HuggingFace page. All models have an architecture similar to BERT but with some extra transformation. The models we selected are: Bart Large, RoBERTa Large, DeBERTa Large and DeBERTa V2 XLarge. For each experiment we analyzed the model predictive performances in terms of accuracy, precision, recall and f1score computed both we respect to the entire dataset and to the fine-grained categories. The accuracy obtained by each model is about 10% lower than the accuracy that the model obtained on the training corpus (in all the cases still quite high). The model that performed better on our dataset is DeBERTa V2 XLarge, which achieved an overall accuracy equal to 81,25%. The same model achieved an accuracy of 91,6% on the MultiNLI.

Finally, we analyzed the specific errors made by this model: the fine-grained category for which the bigger number of errors was produced is the *Restrictivity* (accuracy equal to 69%), while the category over which the model performed better is *Active/Passive* (accuracy equal to 91%). This suggested that just for some of the highlighted phenomena the

model is actually able to grasp well enough the key information for analyzing the entailment relation. Moreover, the predictive capabilities of the model were not satisfying for the label *neutral*: in this case the recall is very low (65%).

These results suggested that, even if pre-trained Transformer models improved a lot the state-of-the-art results for Natural Language Understanding tasks, they are not sufficiently good in grasping information related to some specific cases. In particular, the models analyzed here were not very good at distinguish between restrictive and non-restrictive usage of modifiers and at predicting the entailment relation between pairs of sentences in which the dative case or a coordination scope play a key role. So, it can be useful in the future to better analyze these specific cases with an ad hoc dataset, in which also some rare and complex examples are present.

Bibliography

[1] Lenci A. “Distributional Models of Word Meaning.” In: *Annual Review of Linguistics* (2018), 4:151–171.

[2] Mikolov T. et al. “Efficient Estimation of Word Representations in Vector Space.” In: (2013).

[3] Adi Y. et al. “Fine-grained analysis of sentence embeddings using auxiliary prediction tasks.” In: *International Conference on Learning Representations* (2017), pp. 1–13.

[4] Bengio Y. et al. “A Neural Probabilistic Language Model.” In: *Journal of Machine Learning Research* 3 (2003), pp. 1137–1155.

[5] Bowman S. R. et al. “A Large Annotated Corpus for Learning Natural Language Inference”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (2015), pp. 632–642. DOI: 10.18653/v1/d15-1075.

[6] Conneau A. et al. “Supervised Learning of Universal Sentence Representations from Natural Language Inference Data”. In: *arXiv* (2018). eprint: 1705.02364.

[7] Dagan I. et al. “Recognizing textual entailment: rational, evaluation and approaches”. In: *Natural Language Engineering* 15.4 (2009), pp. i–xvii.

[8] Devlin J. et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: (2019). arXiv: 1810.04805 [cs.CL].

[9] Giulianelli M. et al. “Under the Hood: Using Diagnostic Classifiers to Investigate and Improve how Language Models Track Agreement Information”. In: *arXiv* (2018). eprint: 1808.08079.

[10] He P. et al. “DeBERTa: Decoding-enhanced BERT with Disentangled Attention.” In: *arXiv* (2021).

[11] Kiros R. et al. “Skip-Thought Vectors”. In: *arXiv* (2015). eprint: 1506.06726.

[12] Lewis M. et al. “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.” In: *arXiv* (2019).

[13] Liu X. et al. “Multi-Task Deep Neural Networks for Natural Language Understanding.” In: *arXiv* (2019).

[14] Liu X. et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach.” In: *arXiv* (2019). eprint: 1907.11692.

[15] Peters M. P. et al. “Deep contextualized word representations”. In: *CoRR* abs/1802.05365 (2018). arXiv: 1802.05365.

[16] Radford A. et al. “Improving language understanding by generative pre-training”. In: *OpenAI Blog* (2018).

[17] Rajpurkar P. et al. “SQuAD: 100,000+ Questions for Machine Comprehension of Text”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (2016), pp. 2383–2392. DOI: 10.18653/v1/D16-1264.

[18] Richardson K. et al. “Probing Natural Language Inference Models through Semantic Fragments”. In: *arXiv* (2019), pp. 6085–6090. DOI: 1909.07521.

[19] Tenney I. et al. “What Do You Learn from Context? Probing for Sentence Structure in Contextualized Word Representations”. In: *International Conference on Learning Representations* (2019).

[20] Vaswani A. et al. “Attention Is All You Need.” In: *arXiv* (2017). eprint: 1706.03762.

[21] Wang A. et al. “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding”. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (2018), pp. 353–355. DOI: 10.18653/v1/W18-5446.

[22] XiPeng Q. et al. “Pre-trained models for natural language processing: A survey”. In: *Sci China Tech Sci* 63 (2020), pp. 1872–1897. DOI: https://doi.org/10.1007/s11431-020-1647-3.

[23] Zhang Z. et al. “Semantics-aware BERT for Language Understanding.” In: *arXiv* (2019).

[24] MacCartney B. *Natural language inference*. Stanford University, ProQuest Dissertations Publishing, 2009.

[25] Regier T. Bacon G. “Probing sentence embeddings for structure-dependent tense”. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP* (2018). DOI: 10.18653/v1/W18-5440.

[26] Glickman O. Dagan I. “Probabilistic textual entailment: Generic applied modeling of language variability.” In: *PASCAL workshop on Text Understanding and Mining* (2004).

[27] Magnini B. Dagan I. Glickman O. “The PASCAL Recognising Textual Entailment Challenge”. In: *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment* 3944 (2005), pp. 177–190. DOI: 10.1007/11736790_9.

[28] Resnik P. Ettinger A. Elgohary A. “Probing for semantic evidence of composition by means of simple classification tasks.” In: *RepEval* (2016). DOI: https://doi.org/10.18653/v1/W16-2524.

[29] Pelletier F.J. “The Principle of Semantic Compositionality”. In: *Topoi* 13 (1994), pp. 11–24. DOI: 10.1007/BF00763644.

[30] Alammar J. *The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)*. Last accessed on October 2021. URL: https://jalammar.github.io/illustrated-bert/.

[31] de Marneffe M. Jiang N. “Evaluating BERT for Natural Language Inference: A Case Study on the CommitmentBank”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (2019), pp. 6085–6090. DOI: 10.18653/v1/d19-1630.

[32] Dennis S. Jones M. N. Willits J. *Models of Semantic Memory.* Oxford Handbook of Mathematical and Computational Psychology, 2015, pp. 232–254.

[33] Martin J. H. Jurafsky D. *Speech and Language Processing.* 3rd ed. Stanford University Press, 2021. URL: https://web.stanford.edu/~jurafsky/slp3/.

[34] Erk K. "Vector Space Models of Word Meaning and Phrase Meaning: A Survey." In: *Linguistics and Language Compass* 6(10) (2012), pp. 635–653.

[35] Richardson J. Kudo T. "SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing." In: *arXiv* (2018).

[36] Taylor W. L. ""Cloze Procedure": A New Tool for Measuring Readability". In: *Journalism Quarterly* 30 (4) (1953), pp. 415–433. DOI: https://doi.org/10.1177/107769905303000401.

[37] Tsuchiya M. "Performance Impact Caused by Hidden Bias of Training Data for Recognizing Textual Entailment". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (2018).

[38] Manning C. MacCartney B. "Modeling semantic containment and exclusion in natural language inference." In: *Proceedings of the 22Nd International Conference on Computational Linguistics* 1 (2008), pp. 521–528. DOI: http://dl.acm.org/citation.cfm?id=1599081.1599147.

[39] Lapata M. Mitchell J. "Composition in Distributional Models of Semantics". In: *Cognitive Science: A multidisciplinary journal* (2010).

[40] Manning C. Pennington J. Socher R. "GloVe: Global Vectors for Word Representation." In: 3 (2014), pp. 1532–1543. DOI: 10.3115/v1/D14-1162.

[41] Pal C. Pilault J. El hattami A. "Conditionally Adaptive Multi-Task Learning: improving transfer learning in NLP using fewer parameters less data." In: *arXiv* (2020).

[42] Batiukova O. Pustejovsky J. *The Lexicon*. Cambridge, Cambridge University Press, 2019.

[43] Yu C. T. Salton G. Yang C. S. “A theory of term importance in automatic text analysis.” In: *Journal of the American Society for Information Science* (1975).

[44] Knight K. Shi X. Padhi I. “Does string-based neural MT Learn source syntax?” In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (2016).

[45] Janssen Theo M. V. “Montague semantics”. In: (2006), pp. 244–255. DOI: 10.1016/b0-08-044854-2/01101-9.

[46] Bowman S. R. Williams A. Nangia N. “A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* 1 (2018), pp. 1112–1122. DOI: 10.18653/v1/d15-1075.

[47] Harris Z. “Distributional structure.” In: *Word* (1954), pp. 146–162.