Universidad Politécnica
de Madrid

**Escuela Técnica Superior de
Ingenieros Informáticos**

Master of Science in Artificial Intelligence

Master's Final Project

Automatic Topic Label Generation Using Conversational Models

Author: Virginia Ramón Ferrer
Tutors: Óscar Corcho García, Carlos Badenes Olmedo

Madrid, June 2023

This Master's Thesis has been deposited at the ETSI Informáticos of the Universidad Politécnica de Madrid for its defense.

Master's Final Project
Master of Science in Artificial Intelligence

Title: Automatic Topic Label Generation Using Conversational Models

June 2023

Author: Virginia Ramón Ferrer
Tutors: Óscar Corcho García, Carlos Badenes Olmedo
Department of Artificial Intelligence
ETSI Informáticos
Universidad Politécnica de Madrid

Resumen

La modelización probabilística de tópicos es una técnica de aprendizaje automático no supervisada que, dada un conjunto de documentos, es capaz de analizar, detectar y agrupar automáticamente palabras que mejor caracterizan tópicos comunes presentes en el conjunto. Sin embargo, muchas veces estamos interesados en saber qué relaciona estos documentos más allá de los patrones más característicos o conjuntos de palabras en el conjunto. En consecuencia, surgió la generación de etiquetas de tópicos, que buscaba generar una etiqueta que caracterizara el conjunto de documentos de forma más interpretable que un grupo de palabras que, a priori, no sabemos qué relación tienen entre sí. Actualmente, se siguen investigando nuevas formas de generar estas etiquetas de temas de manera automática y fácilmente comprensibles.

A la vez, recientemente han aparecido Modelos de Lenguaje con fin conversacional, los cuales están entrenados para comprender y generar diálogos entre humanos y máquinas. Estos modelos presentan capacidades más allá de la habilidad de tener una conversación, como por ejemplo ChatGPT, que ha demostrado poder redactar de forma autónoma correos electrónicos o redacciones sobre un tema específico, por ejemplo.

Los modelos conversacionales presentan un potencial aparente no solo para ser aplicados en tareas recreativas, sino que también pueden ser útiles para otras tareas, como se indica en la publicación de Sallam "ChatGPT Utility in Healthcare Education, Research, and Practice: Systematic Review on the Promising Perspectives and Valid Concerns" [1], donde el autor analiza 60 publicaciones que hablaban de los beneficios de usar ChatGPT en diferentes tareas, como el análisis eficiente de conjuntos de datos o la generación de código para la investigación en salud. Ante este hecho, el objetivo de este Proyecto Final de Máster es estudiar la capacidad que pueden tener los modelos conversacionales para generar automáticamente y de manera no supervisada etiquetas para tópicos probabilísticos, dado un conjunto de palabras clave representativas del tema, siguiendo una metodología a la cual nos referiremos como *Etiquetado Conversacional de Tópicos Probabilísticos* (*CPTL o Conversational Probabilistic Topic Labelling en inglés*). También comparamos el rendimiento de estos modelos conversacionales con el rendimiento de un modelo de lenguaje específico para tareas, entrenado para generar etiquetas de temas.

Abstract

Probabilistic topic modelling is an unsupervised machine learning technique that, given a set of documents, is capable of scanning, detecting patterns of words and phrases, and automatically grouping words that best characterize a topic. Many times, however, we are interested in knowing what relates these documents beyond the most characteristic patterns or sets of words in the set. Consequently, the generation of topic labels appeared, which sought to generate a label that would characterize the set of documents in a more interpretable way than having a group of words that we, a priori, do not know the relationship they have with each other. Currently, new ways of generating these topic labels that are easily understandable automatically are still being investigated.

At the same time, Neural Language Models based on Neural Networks with conversational purpose have recently emerged, which are trained to understand and generate dialogues between humans and machines. These models possess capabilities beyond the ability to engage in conversation, such as ChatGPT, which has demonstrated the ability to autonomously compose emails or write about a specific topic, for example.

Conversational models present an apparent potential to not only have recreational applications, but can also be useful for other tasks, as stated in Sallam's publication "ChatGPT Utility in Healthcare Education, Research, and Practice: Systematic Review on the Promising Perspectives and Valid Concerns" [1], where the author analysed 60 publications that talked about the benefits of using ChatGPT in different tasks, such as efficient analysis of datasets or code generation for health care research. Given this fact, the purpose of this Final Master's Project is to study the capacity that conversational models may have to automatically and unsupervisedly generate tags for probabilistic topics given a set of keywords representative of the topic, following a methodology which we will refer as *Conversational Probabilistic Topic Labelling* (*CPTL*). We also compare the performance of these conversational models with the performance of a task-specific language model trained to generate topic labels.

Contents

Chapter 1

Introduction

Probabilistic topic modelling is a powerful and widely used unsupervised machine learning technique that, when provided with a collection of documents, such as articles or social media posts, has the ability to analyse and detects recurring patterns of words and phrases that represent the topics present in them. The objective of topic modelling is to automatically identify and group together words and expressions that are closely related and to capture the underlying topics within a document collection or corpus. It employs a probabilistic approach, which applies statistical models to estimate the likelihood of certain words appearing together in a given topic. By analysing co-occurrence patterns over word and phrases probabilities, the model identifies words that frequently appear together and infers the topic or theme they represent. The resulting output of probabilistic topic modelling is a set of groups, or clusters, of words that best characterize the topics, where each cluster represents a specific topic. These topic models can then be used for various purposes, such as document classification, information retrieval and recommendation systems, among others.

We stated that probabilistic topic modelling groups words together to define a pattern that represents a topic, but these words are grouped without a clear understanding of their interrelationships. For example, imagine that a group of people is given the set of words "*america, continent, asia, population, include, ocean, language, africa, region, people*" and is asked to infer the possible label to give to this topic only taking into account these top 10 words. Each of them start guessing and proposing different labels as "geography", "the earth" or "continent", for example, but how do they choose which one of these labels is the most fitting? They have to try and reach an agreement, but this discussion process is time-consuming and may not lead to an agreement. Now imagine that they are given hundreds or thousands of sets of words and they have to generate the labels for each of them, it is obvious that it may not be feasible to do this "by hand" and they will need a more efficient method to generate these labels. Given this fact, efforts have been made, and still are, to generate topic labels that could better represent the topics in the document collection. The generation of topic labels seeks to create labels that can convey the overarching theme or concept represented by a group of related words. However, generating topic labels that are both accurate and easily understandable is a challenging task. It requires extracting the key semantic elements from the words and phrases within a topic cluster and finding an appropriate label that captures the essence of those elements. Ongoing research is focused on exploring new methods for automatically generating topic labels that can automatically analyse the content and context of a topic cluster, identify salient keywords and phrases, and synthesize them into coherent and interpretable topic labels. While significant progress has been made in generating topic labels for probabilistic topic modelling, there is still

ongoing research and exploration into novel approaches, where the objective is to enhance the clarity of the outcomes and create automated methods capable of producing informative and comprehensible topic labels. This task will aid in extracting valuable insights from an extensive document corpus.

As new technologies, particularly the Internet, have emerged, society has entered an era of producing vast volumes of data on a daily basis. This data is accessible to anyone and includes various forms of text, ranging from scientific articles and news pieces to brief messages like a mere "Tweet." Consequently, there is a growing demand for faster and automated methods of generating topic labels without relying on human intervention. Coinciding with this trend, language models have recently emerged with the aim of accurately representing and understanding natural language. These models are statistical models that are used in natural language processing *(NLP)* to predict the probability of a sequence of words. Currently, these models find extensive application in various fields of *NLP*, encompassing tasks such as machine translation, text generation, and information retrieval. Particularly, advanced language models based on neural networks present the ability to detect intricate patterns and deliver more precise outcomes when compared to other simpler models. Among these more complex models are conversational models, that are specifically designed to comprehend and generate natural dialogues between humans and machines. Unlike conventional language models that primarily focus on predicting the next word in a text sequence, conversational language models consider the whole context of an entire conversation. To do this, these models apply techniques such as long-term memory, attention and user feedback to better understand the intentions and needs of the user to generate relevant responses. Conversational models present great potential beyond merely conversational tasks. A very clear example of the capacity of these models can be ChatGPT, which not only has the capacity to hold a conversation, but can also carry out tasks of another nature, such as generating programming code or creating poems. Currently, the generation capacity of these models is not known, but the potential they have can be clearly seen.

Given the previous context, we decided to further explore the capacity of conversational models while trying to answer a series of research questions:

- **RQ1** $\rightarrow$ *How can conversational models be used to generate labels for probabilistic topics?*
- **RQ2** $\rightarrow$ *How can topic labels be extracted when given a conversational model's answer?*
- **RQ3** $\rightarrow$ *What is the quality of the topic labels generated by the conversational models?*
- **RQ4** $\rightarrow$ *What is the most adequate number of words to describe a probabilistic topic?*

The goal of this Final Master's Project was to explore how conversational models can automatically and without supervision generate tags or labels for topics by answering the research questions exposed. We wanted to understand their ability to do this by providing a set of representative keywords for each topic. Additionally, we compared the performance of these conversational models with a task-specific language model that was trained specifically to generate topic labels. We proposed a system to conduct a limited study on some of the most outstanding publicly available conversational models. We will refer to our method as *Conversational Probabilistic Topic Labelling* (*CPTL*).

Along this document we review the related work already done about probabilistic topic modelling, topic label generation and language models in chapter 2. Then we introduce the approach adopted to solve the generation of topic labels using conversational models, presented in chapter 3, and the evaluation methodology followed along the experiments, presented in

chapter 4. Finally, we exposed the results and the analysis of our experiments in chapter 5 and the conclusions and future work extracted from this process in chapter 6.

Chapter 2

Related work

Topic Label Generation is a subfield of natural language processing (NLP) that involves automatically generating human-interpretable labels for topics identified by topic modelling algorithms. The goal of the generation of topic labels is to provide concise and informative labels that summarize the main concept of each topic, thus facilitating the interpretation and understanding of large collections of documents.

Along this chapter we present a general view of the Probabilistic Topic Modelling and Topic Label Generation concepts and related work, specifically presented in sections 2.1 and 2.2 respectively. We will also further present the concept of Language Models, specifically conversational models, and analyse their capabilities in section 2.3.

2.1 Probabilistic topic modelling

Topic modelling algorithms are statistical methods that analyse the words of original texts to discover the topics that run through them, how these topics are connected to each other and how they change over time [2]. They do not require any prior annotations or labelling of the documents, that is, the topics are found from the analysis of the original texts. Probabilistic Topic modelling, then, is a sub-field of Natural Language Processing (NLP) that aims to automatically identify and extract topics from a large corpus of documents without prior information about the nature of these topics. Typically, these topics are expressed as a collection of the most representative words that characterize a particular topic.

There are several approaches to probabilistic topic modelling, but the most widely known are probably *Probabilistic Latent Semantic Indexing* (*PLSI*) [3], *Latent Dirichlet Allocation* (*LDA*) [4] and its extensions. ***Probabilistic Latent Semantic Indexing* (*PLSI*)**, also known as *Probabilistic latent semantic analysis (PLSA)*, first introduced in 1999, is based on the idea of representing words and documents as probability distributions over latent topics, assuming that each document in a corpus is generated from a mixture of topics, and that each word in the document is generated from one of those topics. The model learns the probability distributions of the topics and the likelihood of each word given each topic. *PLSI* cannot handle new words or documents that were not present in the training corpus, which led to the development of more advanced models such as *Latent Dirichlet Allocation* (*LDA*), which is an extension of *PLSI* that can handle new words and documents.

Topic ID	1									
Words	comments	read	nice	post	great	april	blog	march	june	css
Probability	0.015	0.010	0.050	0.032	0.026	0.021	0.019	0.011	0.008	0.004

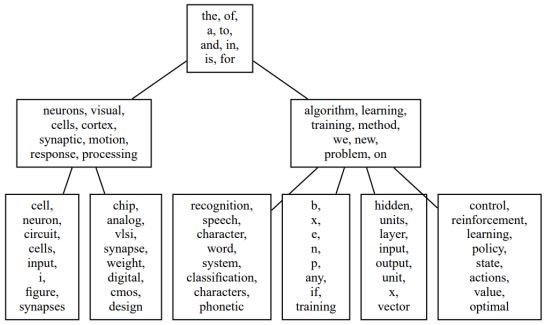
Figure 2.1: PLSI topics' representation

***Latent Dirichlet Allocation* (*LDA*)**, first introduced in 2001, is a generative probabilistic model of a corpus that is based on the idea that documents are represented as random mixtures over latent topics, having that the number of possible topics is known and fixed, where each topic is characterized by a distribution over a number of words also known. By estimating the distribution of topics in each document and the distribution of words in each topic *LDA* can uncover the underlying themes or topics in a corpus. This model is restricted by the assumption that the number of topics and words in each topic is always fixed, having that this assumption may not always be true. Also, *LDA* assumes that the order of the documents or the order of the words in a document is not relevant. Furthermore, it also assumes that the topics are static and do not change over time.

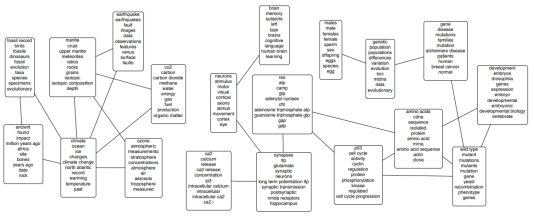
Topic ID	1									
Words	comments	read	nice	post	great	april	blog	march	june	css
Probability	0.015	0.010	0.050	0.032	0.026	0.021	0.019	0.011	0.008	0.004

Figure 2.2: LDA topics' representation

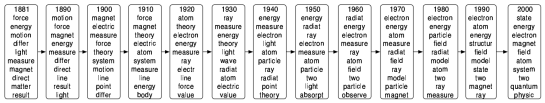
Over the years, several extensions and variations of *LDA* have been proposed to improve its performance and address some of its limitations. In 2003, the ***Hierarchical LDA* (*hLDA*)** [5] was proposed. *hLDA* extends *LDA* by modelling topics as a hierarchy where each topic is a subtopic of a more general topic, relaxing the assumption that the number of topics has to be fixed. It uses a nested version of the Chinese restaurant process [6], a process based on the idea of customers arriving at a restaurant and sitting at tables, with each table representing a cluster or group of customers. Two years later, in 2005, the ***Correlated Topic Models* (*CTM*)** [7] were presented as an extension of *LDA*, that, instead of assuming that each document is generated independently from a mixture of topics, assume that the topics are correlated across documents, which means that the same set of topics can appear in multiple documents, but their prevalence can vary depending on the document. In 2006, the ***Dynamic Topic Models* (*DTM*)** [8] were introduced as an extension of *LDA* that models the topic evolution over time. They assume that the distribution of topics in a document can change over time, and allows the discovery of new topics and the fading away of old ones. Three years later, in 2009, ***Labelled LDA*** (*L-LDA*) [9] was presented as a topic model that constrains Latent Dirichlet Allocation by defining a one-to-one correspondence between LDA's latent topics and previously known user tags.



(a) *hLDA* [5]



(b) *CTM* [7]



(c) *DTM* [8]

Topic Label	Web									
Words	web	search	site	blog	css	content	google	list	page	posted
Probability	0.015	0.010	0.050	0.032	0.026	0.021	0.019	0.011	0.008	0.004

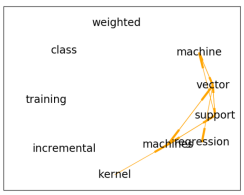
(d) *L-LDA*

Figure 2.3: LDA-based topic modelling methods topics' representation

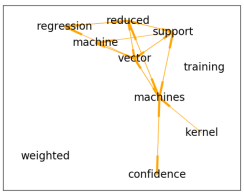
There are others approaches to topic modelling that do not rely on the *LDA* assumptions, such as ***Biterm Topic Models* (*BTM*)** [10], that was proposed in 2014 as a novel way for short text topic modelling, derived from *LDA*. *BTM* learns topics by directly modelling the generation of word co-occurrence patterns in the corpus, making the inference effective with the rich corpus-level information. This algorithm first creates a vocabulary of all unique biterms in the corpus and assigns a probability distribution over topics to each of them, where a topic is defined as a set of related biterms. Finally, it infers the underlying topic structure of the corpus based on the distribution of biterms. *BTM* is able to capture the co-occurrence patterns of words and detect latent topics that are not explicitly represented in the corpus. On the other had, ***Non-negative Matrix Factorization (NMF)*** [11] is an alternative to *LDA* that uses a matrix factorization approach to identify topics. *NMF* assumes that each document is a combination of a small number of topics and that the words in each document are generated from a linear combination of those topics, given that the documents are represented in a latent semantic space derived by *NMF*, where each axis captures the base topic of a particular document cluster, and each document is represented as an additive combination of the base topics. More recently, in 2018, Shi et al. [12] proposed a ***Semantics-assisted Non-negative Matrix Factorization* (SeaNMF)** model to discover topics for the short texts, where they effectively incorporated word-context semantic correlations into the model that were learned from the skip-gram view of the corpus.

Topic ID	1									
Words	comments	read	nice	post	great	april	blog	march	june	css
Probability	0.015	0.010	0.050	0.032	0.026	0.021	0.019	0.011	0.008	0.004

(a) *BTM*



(b) *NMF* [12]



(c) *SeaNMF* [12]

Figure 2.4: Non-LDA based topic modelling methods topics' representation

2.2 Topic label generation

Topics are normally represented as a subset of the most relevant words in that topic (figure 2.2), but these representations have a limited representative capacity. Another approach is to manually generate labels, as Mei et al. [13] did in their "A probabilistic approach to spatio-temporal theme pattern mining on weblogs" publication in 2006. Some approaches combine manual and automatic methods. In 2016, for example, Atapattu and Falkner [14] proposed a framework for generating and labelling topics from discussions in Massive Open Online Courses (MOOCs). The authors used a combination of automatic and manual methods to label the topics. The automatic method involved using the most probable words generated by LDA to create an initial set of labels. The manual method involved a group of human annotators reviewing the initial labels and refining them to ensure they were meaningful and accurate. Obviously, this manual approach is not scalable and introduces a bias, as human input is the main source of information.

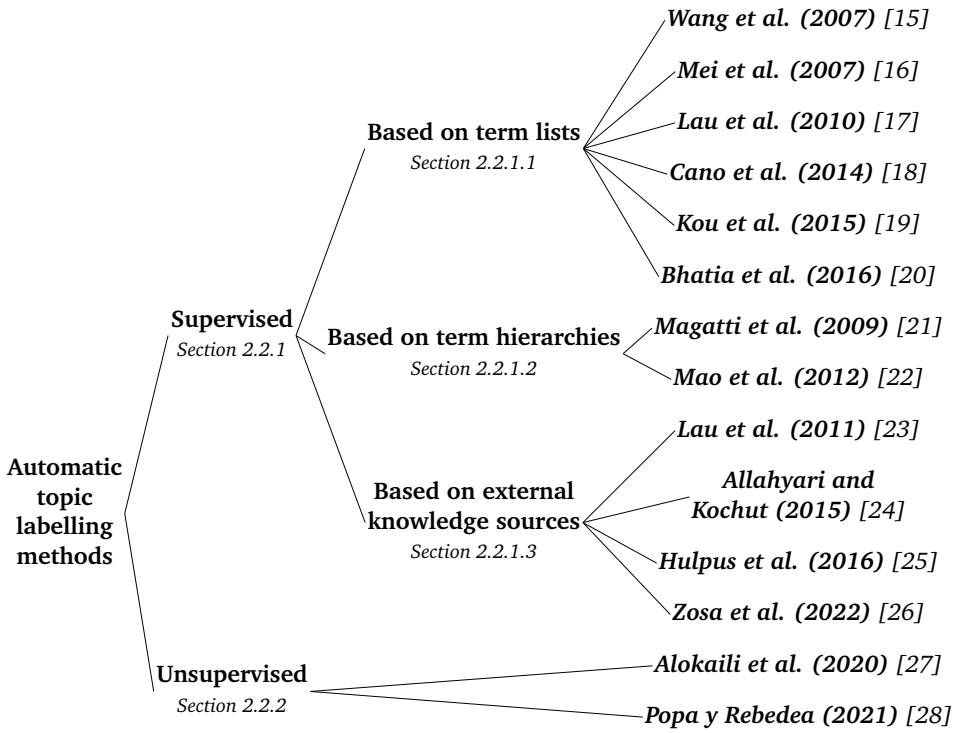


Figure 2.5: Automatic topic labelling methods

In the last few years automatic label generation has been gaining attention in research. Along this chapter we present different approaches to automatic label generation. As presented in figure 2.5, we divided the approaches in "supervised" and "unsupervised" methods, in sections 2.2.1 and 2.2.2. Inside the "supervised" methods we can also find methods based on term lists, in section 2.2.1.1, based on term hierarchies, in section 2.2.1.2, and based on external knowledge sources, in section 2.2.1.3.

2.2.1 Supervised methods

We understand as "supervised" methods as methods that have a defined selection of possible labels to assign to the topics when assigning the topic labels. In this section we can see that the methods use different techniques that seek to find the best fitting label.

Along this section we present methods based on term lists, in section 2.2.1.1, based on topic hierarchies, presented in section 2.2.1.2, and methods that employ external knowledge sources as label candidates, explained in section 2.2.1.3.

2.2.1.1 Term lists

In 2007 Wang et al. presented [15] a topical n-gram model (TNG) that automatically determined uni-gram words and phrases based on context and assigned a mixture of topics to both individual words and n-gram phrases, where the top n-grams of a topic could be used as topic labels.

Concurrently, Mei et al. [16] proposed an automatic topic labelling approach that converted the labelling problem to an optimization problem that sought the minimum Kullback-Leibler (KL) divergence and the maximum mutual information with the topic of the candidate labels. This method can be applied to labelling a topic generated by different topic modelling approaches, as PLSA and LDA, for example.

In 2010, Lau et al. [17] presented a labelling method based around the assumption that an appropriate label for a topic can be found among the high-ranking terms in a topic model. They assessed the suitability of each term by way of comparison with other high-ranking terms in that same topic, using simple pointwise mutual information and conditional probabilities. They experimented both with a simple ranking method based on the component scores and a ranking support vector regression (SVR) framework where they used the component scores along with features from WordNet and from the original topic model.

In 2014, Cano et al. [18] addressed the problem of automatic labelling of latent topics learned from Twitter as a summarization problem. They based their method on term relevance of documents related to the topics and the use of summarization algorithms. Specifically they investigated the use of lexical features by comparing three different well-known multi-document summarization algorithms against the top-n topic terms baseline, these being *Sum Basic (SB)*, *Hybrid Term frequency – Inverse Document Frequency (Hybrid TF-IDF)*, *Maximal Marginal Relevance (MMR)* and *Text Rank (TR)*. During the evaluation, they compared the results of the summarization techniques with the *Top Terms (TT)* of a topic as our baseline and concluded that, in general, these techniques outperformed *TT*, especially *SB* and *Hybrid TF-IDF*.

In 2015, Kou et al. [19] proposed a topic labelling framework that used word vectors, specifically Skip-gram, Continuous Bag of Words (CBOW), and tri-gram vectors. They first used a chunk parser to generate a set of candidate labels identifying topic-related document sets and extract chunks of them that contained words from the top-10 words of the topic to be used as candidate labels. Then they mapped topics and candidate labels to word vectors and letter tri-gram vectors in order to find which candidate label was more semantically related to that topic, using the similarity between a topic and its candidate label vectors to find the topic labels.

In 2016, Bhatia et al. [20] proposed NETL, an approach to topic labelling based on word and document embeddings, which both automatically generated label candidates given a topic input, and ranked the candidates in either an unsupervised or supervised manner, to produce the final topic label. Following a similar structure as Lau et al. [23], they first generated candidate topic labels based on English Wikipedia and then ranked these topic labels. In this case, they generated the embeddings based on Wikipedia entries which were subsequently compared to the topic embedding and ranked by similarity. For each embedding based on Wikipedia entries, its entry title were used as the candidate topic labels.

2.2.1.2 Term hierarchies

In 2009, Magatti et al. [21] presented an algorithm for the automatic labelling of topics accordingly to a topic hierarchy, implemented through a tree where it is assumed that the available labelling schema is summarized, to find the optimal label according to a set of similarity measures and a set of topic labelling rules. The labelling rules are specifically designed to find the most agreed labels between the given topic and the hierarchy. The hierarchy is obtained from a document corpus obtained from the Google Directory (gDir) service, extracted via an ad-hoc developed software procedure and expanded through the use of the OpenOffice English Thesaurus, which was used to obtained a thesaurized topics tree using WordNet.

In 2012, Mao et al. [22] proposed two algorithms that automatically assigned concise labels to topics in a hierarchy by exploiting sibling and parent-child relations among topics. Given topic models, for each topic, this algorithm generated a set of candidate phrases by extracting noun phrases and verb phrases that were highly associated with the topic. Then, they ranked the candidate labels by exploiting the structural relation among topics to find the most fitting candidate label. To do this they proposed the use of two ranking methods: Term Weighting Based Ranking (TWL), where they used global term weighting schemes, and Statistical Significance Based Ranking, where they used comparative statistics like Jensen-Shannon Divergence (JSD).

2.2.1.3 External knowledge sources

In 2011, Lau et al. [23] proposed an automatic topic labelling method that, based on LDA topics, sourced topic label candidates from Wikipedia by querying with the top-N topic terms, identified the top-ranked document titles; and post-processed the document titles to extract sub-strings. First, they mapped the topic to a set of concepts by querying Wikipedia, using the top-10 topic terms and the top-8 entry titles that were selected as primary candidate topic labels. Secondary labels were generated from component n-grams contained within the primary candidates, and filtered out incoherent and unrelated titles measuring their similarity with the primary labels, based on Wikipedia document categories. Finally, the combined the set of primary and secondary label candidates was ranked using a number of lexical association features, either directly in an unsupervised manner or indirectly based on training a support vector regression model.

In 2015, Allahyari and Kochut [24] proposed a method for automatically labelling topics generated from topic models using domain-specific ontologies. The proposed topic modelling approach, called *OntoLDA*, combined a standard topic modelling algorithm, *LDA*, with an ontology-based process to generate more accurate and interpretable topics and labels. The approach used an existing domain-specific ontology to identify and extract relevant terms and concepts related to each topic. They based their method on the intuition that entities occurring in the text and their relationships can determine the topic related, so they relayed on the semantic similarity between the text and fragments of the ontology to identify the possible topics. The authors, once they had identified the topic, used the ontology concepts and their hierarchy to generate the topic labels constructing a semantic graph from the top concepts related to the topic, followed by the selection of a sub-graph of this graph to define a themantic graph from which a topic graph will be extracted and, finally, extracted the top labels from the topic label graph given the semantic similarity between the topic and the candidate labels.

In 2016 Hulpus et al. [25] proposed an automatic topic labelling approach by exploiting structured data from DBpedia, a project aiming to extract structured content from the information created in the Wikipedia project. They based their approach on the hypothesis that words co-

occuring in a text likely refer to concepts that belong closely together in the DBpedia graph. Given a topic, they first found the terms with highest marginal probabilities, and then determined a set of DBpedia concepts where each concept represents the identified sense of one of the top terms of the topic. After that, they created a graph out of the concepts and use graph centrality algorithms to identify the most representative concepts for the topic.

In 2022, Zosa et al. [26] proposed an ontological mapping method that mapped topics to concepts in a language-agnostic news ontology, which concepts had labels in multiple languages that were used as topic labels. The authors treated the ontology mapping problem as a multi-label classification task where a topic, described its top-n terms, could be classified as belonging to one or more concepts in the ontology. They used the Sentence-BERT [29], a modification of the pre-trained BERT network that use siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity, to encode the top-n terms to subsequently carry out the classification.

2.2.2 Unsupervised methods

All the approaches presented along section 2.2.1 share the dependency on a previous selection of possible labels, whether it comes in the form of a list of terms, a given terms' hierarchy or a wider knowledge database. The need for prior knowledge adds a factor of bias and limitation of generalization given the restrictions imposed by the data set that is used as possible labels. It also restricts the scalability and portability of the method to the knowledge sources that are available in relation to the desired task to complete. In this section we present a number of approaches that aim to minimize the restrictions in generalization and scalability, and the bias factor by avoiding the dependency on previous selections of possible labels. We understand as "unsupervised" methods as methods that have do not have a defined selection of possible labels to assign to the topics when assigning the topic labels. In this case, we mainly have methods based on contextualized embeddings.

In 2020, Alokaili et al. [27] presented an neural-based model that automatically generates labels for topics in a single step, instead of retrieving and ranking candidates as done in previous explained methods. The authors proposed a sequence to-sequence RNN-based encoder-decoder architecture trained with distant supervision using Wikipedia page titles and BERTScore [30], an automatic evaluation metric for text generation based on pre-trained BERT contextual embeddings, to evaluate the quality of the generated labels. To train their model, they generated two datasets by selecting pairs of titles and articles from Wikipedia, one where the titles are treated as the labels and the top 30 words from each article ranked by TF-IDF are treated as synthetic topic terms, and the second where the first 30 words from the article are used as topic terms.

In 2021, Popa y Rebedea [28] proposed a method for automatically generating topic labels from a collection of documents called *BART-TL*. The method is based on the BART (Bidirectional and Auto-Regressive Transformer) model, which is a large-scale language model that has been pre-trained on a large corpus of text. They leveraged generative transformers to learn accurate representations of the most important topic terms and candidate labels. This was achieved by fine-tuning pre-trained BART models on a large number of potential labels generated by state of the art non-neural models for topic labelling, enriched with different techniques. Specifically they fine-tuned two models: *BART-TL-ng*, which was fine-tuned with a baseline dataset generated from the NETL labeller [20] in addition to space-separated n-grams sampled from the most important words in the topic, and *BART-TL-all*, which, in addition to the baseline dataset and the n-grams, was also trained with groups of sentences and noun phrases from the corpus.

Our approach, *Conversational Probabilistic Topic Labelling* (*CPTL*), would be located in this section, as we use a non-supervised approximation based on conversational models to generate the topic labels. Our approach differs from the previous ones from the type of language model used, as we do not specifically train the models to carry out the topic labelling but instead use a more general purpose kind of models that are already trained and do not need to be fine-tuned, which enables our methodology to be executed using a wider range of models.

2.3 Language Models

In the world of artificial intelligence, language models play a vital role in understanding and creating human-like language. These algorithms have the ability to unravel the intricacies of language, allowing them to grasp, anticipate, and produce sentences that make sense and fit a given context. Language models are sophisticated algorithms that have been trained on extensive amounts of text data, including books, articles, and websites, through which they gain knowledge of the patterns, structures, and meanings that form the foundation of human language. Language models have been applied to a wide range of tasks, from helping to complete sentences and translate languages to analysing sentiments and generating content. Their versatility and adaptability have made them essential in fields like natural language processing, virtual assistants, and chatbots, revolutionizing how we interact with language. We are focused on conversational models, that are language models specially trained to engage in human-like conversations. They use natural language processing and machine learning techniques to understand user inputs and generate relevant and coherent responses. These models are trained on large datasets of conversational data to learn patterns, context, and semantic relationships in language. We also briefly brush over Question-answering models, that are language models designed to understand and respond to questions by extracting relevant information from a given context or document. They provide accurate and relevant answers to specific queries in natural language.

Language models have been in development for a long time, but with the appearance of new technologies, they have recently experimented a boom in their development. In 1996 Weizenbaum presented ***ELIZA*** [31]. Eliza was a computer program created at MIT that simulated a conversation with users, emulating a Rogerian psychotherapist. Eliza used basic pattern-matching techniques to ask questions and provide responses, often reflecting the user's words back to them. While its purpose was not to genuinely understand or offer therapy, Eliza sparked interest in natural language processing and chatbot development and its popularity contributed to advancements in conversational agents, paving the way for more sophisticated chatbots and virtual assistants in the field.

In 1970 Winograd presented ***SHRDLU*** [32] as a groundbreaking natural language understanding program. It introduced a virtual world where users could interact with blocks through language instructions. By understanding and interpreting user inputs, *SHRDLU* demonstrated early advancements in language processing and showcased the potential of human-computer interaction.

In 1993, ***IBM Model 1*** [33], also known as the IBM Alignment Model, was presented by Brown et al. as part of the IBM Statistical Machine Translation (SMT) project. It was a work in statistical machine translation and introduced the notion of word alignment using an expectation-maximization algorithm. *IBM Model 1* focused on aligning words between a source language and a target language, laying the foundation for subsequent models in the IBM series. *IBM Model 1* made significant advancements by enabling the automatic extraction of word align-

TIMELINE 1: *Language Models*

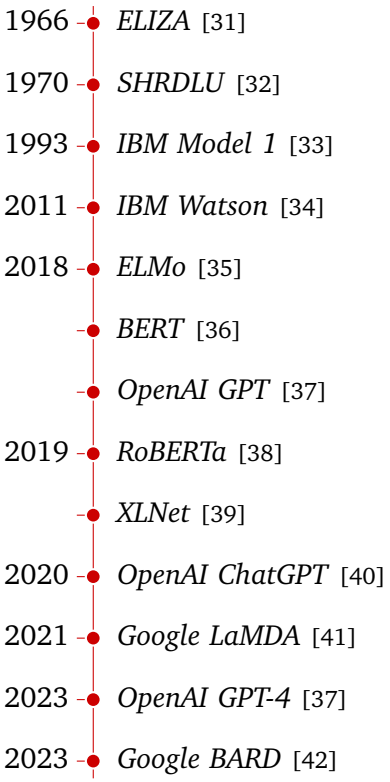


Figure 2.6: Language models timeline

ments from parallel corpora, which helped in improving the accuracy of machine translation systems. The model was later refined and extended with subsequent versions like *IBM Model 2* and *IBM Model 3*, leading to further advancements in statistical machine translation techniques.

In 2011 ***Watson*** [34], created by Ferrucci et al., was first presented to the public through an appearance on the game show Jeopardy! by IBM. *Watson* utilized a combination of advanced techniques, including machine learning, question-answering, and natural language understanding, to compete against human contestants in a complex trivia game. *Watson*'s success on Jeopardy! demonstrated its ability to process and understand natural language queries, analyse vast amounts of information, and generate accurate and contextually relevant answers. The presentation of *Watson* showcased the potential of AI in the domain of question answering and paved the way for further advancements in cognitive computing.

ELMo *(Embeddings from Language Models)* [35] was presented in 2018 by Peters et al. Unlike traditional word embeddings that provide fixed representations for words, *ELMo* generated word representations that captured contextual information based on the surrounding words in a sentence. This was achieved using a bidirectional language model trained on a large corpus. The *ELMo* embeddings demonstrated improved performance in various natural language processing tasks, such as sentiment analysis and question answering, by capturing the nuances of word meaning in different contexts. *ELMo* advanced the field by highlighting the significance of contextualized word embeddings in language understanding and downstream applications.

BERT *(Bidirectional Encoder Representations from Transformers)* [36] was also presented in 2018 by Devlin et al. as part of the Google AI Language team. BERT introduced a breakthrough in

language representation by pre-training a deep bidirectional transformer model on a massive amount of unlabelled text data. The model leverages both left and right context during training, allowing it to capture rich contextual information. *BERT*'s pre-training is followed by fine-tuning on specific downstream tasks, enabling it to excel in various natural language processing tasks such as text classification, named entity recognition, and question answering. *BERT*'s advancements in contextualized word representations and transfer learning significantly improved language understanding models, opening up new possibilities in natural language processing and achieving state-of-the-art performance on several benchmarks. In 2019 ***RoBERTa*** [38], an optimized *BERT* model, was presented by Yinhan Liu et al. as part of the Facebook AI Team. RoBERTa introduced several modifications to the BERT training methodology, including larger training data, longer training duration, and dynamically changing the masking pattern during pre-training, which led to enhanced language representation and improved performance on various downstream tasks.

XLNet *(eXtreme Language understanding Network)* [39] was presented in 2019 by Yang et al. *XLNet* introduced a novel pre-training approach that addressed the limitations of traditional autoregressive models like *BERT*. It leveraged permutation-based training, allowing each token to attend to any other token in a given sequence, enabling bidirectional context learning while avoiding the inconsistency of *BERT*'s masked language modelling. This approach achieved state-of-the-art results on various language understanding benchmarks by capturing bidirectional dependencies and effectively modelling contextual relationships. *XLNet* demonstrated the importance of context modelling in language understanding and advanced the field of pre-training techniques for natural language processing tasks.

OpenAI's ***GPT*** *(Generative Pre-trained Transformer)* [37] was first presented in 2018 by Radford et al. GPT introduced a novel approach to language modelling by pre-training a deep neural network on a massive amount of text data and fine-tuning it for specific downstream tasks. This unsupervised pre-training followed by supervised fine-tuning enabled *GPT* to learn rich representations of language and exhibit impressive capabilities in tasks such as text generation, translation, and comprehension. The subsequent advancements in *GPT* models, such as *GPT-2, GPT-3* and *GPT-4* increased the model size, training data, and computational power, resulting in significant improvements in language understanding and generation, pushing the boundaries of natural language processing and AI applications. ***ChatGPT*** [40] was presented in 2020 as an extension of the *GPT-3* model with a specific focus on generating conversational responses. Simultaneously with the *GPT* models' progress, *ChatGPT* is being refined to enhance its conversational abilities while also incorporating ethical considerations and safety protocols to mitigate potential risks associated with large-scale language models.

Google LaMDA *(Language Model for Dialogue Applications)* [41] was announced in 2021 during the Google I/O conference. *LaMDA* represents a breakthrough in conversational AI and language modelling, focusing on enhancing the capabilities of dialogue-based applications by improving language understanding and generating more natural and coherent responses. *LaMDA* was presented as a model trained on a vast corpus of dialogue data, enabling it to engage in more meaningful and contextually rich conversations. It aims to address challenges such as maintaining longer and more coherent interactions, understanding nuanced queries, and providing more accurate responses. In 2023 Google introduces ***BARD*** [42], an experimental conversational AI service, powered by a lightweight model version of *LaMDA*.

Our proposal, *Conversational Probabilistic Topic Labelling* (*CPTL*), makes use of conversational models to provide the topic labels in an answer and Question-Answering models to extract the topic labels from the conversational models' answers.

Chapter 3

Approach

We explored the possibility of generating probabilistic topic labels using conversational models, which we refer as *Conversational Probabilistic Topic Labelling* (*CPTL*). In section 2.3 we talked about complex language models and their performance in not only conversational tasks, but other tasks also. This motivated us to research their capability of generating labels for probabilistic topic models, as nowadays there still isn't an optimal method to automatically generate topic labels. Along section 3.1 we answer the research questions ***RQ1***, "*How can we use conversational models to generate labels for probabilistic topics?*", and ***RQ2***, "*How can we extract the topic label given the conversational model's answer?*".

3.1 Topic Labelling system based on Conversational Models

A conversation is an interactive and verbal exchange of information, ideas, thoughts, or emotions between two or more individuals that involves a back-and-forth flow of communication where participants take turns speaking and listening to one another. When having a conversation with a machine, something similar happens. A conversational model is trained to understand user inputs, generate relevant and coherent responses, and maintain a smooth flow of dialogue similar to a human conversation. During a conversation we can ask for information of some type and, if the other participants understand the question and know the answer, we can receive this information. We want the conversational model, that in this case acts as the other participant of the conversation, to give us a specific information, a topic label. To receive this label, we have to provide the necessary information not only for it to identify that we want a topic label, but which topic's label we want, that is, we have to indicate that we want to recieve a topic label and the topic we want the label for. Given this, we know that we need to define a question to ask the conversational model that requests for the label of a topic represented as a set of words, given that we will work with probabilistic topic models. We chose to use top words because it offers simplicity and interpretability, and also reduces the dimensionality of the data, as it only needs to store a few words to represent each topic, making computations more efficient. We also decided to compare the results given the whole answer of the conversational models and a "reduced" version of this answer generated by entering this answer to a Question-Answering Model that is asked to identify the main topic of the conversational answer. In figure 3.1 we represented the four-stage pipeline that we defined to generate the labels:

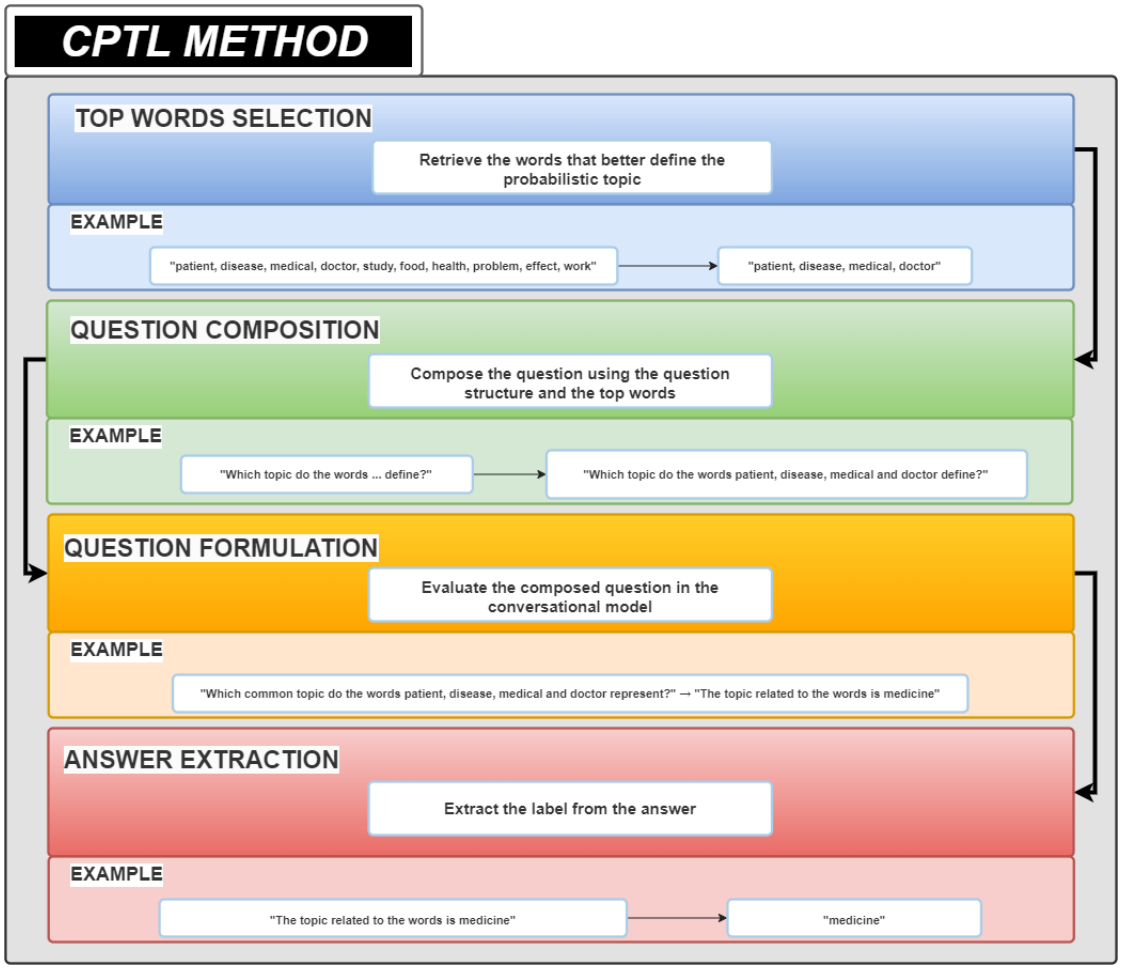


Figure 3.1: Approach structure and example

1. **Top words selection** → We retrieve the words that best represent the probabilistic topic.
    - ***Execution*** → Given the words that represent the probabilistic topic, we select the top n most relevant words of the topic, being n a defined value determined from an evaluation over a range of alternatives.
    - ***Example*** → Given a topic defined by the set of words "patient, disease, medical, doctor, study, food, health, problem, effect, work". We determined that the first four words are the ones that best encapsulate the meaning of the topic, so we retrieve the words "patient, disease, medical, doctor" as candidates to be used in question composition.
2. **Question composition** → We generate the question to ask to the conversational model.
    - ***Execution*** → To generate the question we use a predefined question structure q, where q is a defined question structure determined from an evaluation over a selection of predefined question templates. The question is constructed using the question template q and the top n most relevant words of the topic.
    - ***Example*** → We define that we will ask the model a question with the structure "*Which topic do the words ... define?*". Given this structure and the words selected, our question is "*Which topic do the words patient, disease, medical and doctor define?*".

3. **Question formulation** → We use a conversational model to evaluate the question generated in the previous step as input.
    - ***Execution*** → The question constructed is given to a conversational model as input. The conversational model then generates an answer in which, theoretically, the topic label is contained.
    - ***Example*** → In the previous example, we evaluate the question "*Which topic do the words patient, disease, medical and doctor define?*" as input to a conversational model and we receive the answer "*The topic related to the words is medicine*".
4. **Label extraction** → The answers generated by the conversational model can have more information aside from the label we want to obtain, so we extract the label from the answer.
    - ***Execution*** → Given the conversational model's prediction, we attempt to identify the label given in the answer and extract it from the context to keep exclusively the simplest form of the label of the probabilistic topic. To extract the label, we use Question-Answering language models given their capacity to identify and extract information given a question and a context.
    - ***Example*** → Given the received answer "*The topic related to the words is medicine*", we are only interested in the label "*medicine*", so we extract this label from the whole answer and keep this label as our topic label.

Chapter 4

Evaluation

Along this chapter we explain the evaluation process followed to analyse *CPTL* on the *LibrAIry*'s 20 newsgroups topic model [43], where we seek to answer the research question ***RQ3***, "*What is the quality of the topic labels generated by the conversational models?*". 20 newsgroup dataset comprises around 20.000 newsgroups posts on 20 topics. As we can see in table 4.1, in *LibrAIry*'s 20 newsgroups topic model there are 20 labelled topics in the column "Name" described by 10 relevant words that represent these topics, shown in column "Description". This words were obtained applying LDA [4] and Labelled-LDA [9] (presented in section 2) to the 20newsgroup dataset corpus using the *LibrAIry* framework [43]. Our main goal was to evaluate the capacity of the conversational models to generate topic labels given a subset of these words. In our case, we evaluate questions of the style "The words 'game' 'team' 'play' 'hockey' 'player' are related to which common topic?". We then evaluated the answers received for the topic "sport hockey". After the evaluation of our method proposed, we compared our results to those generated by a task-specific trained model, BART-TL [28].

Id	Name	Description
0	sport hockey	game,team,play,hockey,player,win,goal,season,fan,playoff
1	religion atheism	god,religion,atheist,moral,claim,point,objective,good,belief,argument
2	science space	space,nasa,launch,system,orbit,earth,mission,satellite,shuttle,moon
3	science medicine	patient,disease,medical,doctor,study,food,health,problem,effect,work
4	politics_misc	government,president,state,law,work,give,man,american,drug,stephanopoulo
5	computer mac hardware	mac,apple,problem,drive,system,work,monitor,computer,card,disk
6	politics mideast	armenian,israel,turkish,jew,arab,israeli,muslim,state,kill,government
7	computer ibm hardware	drive,card,system,problem,work,controller,disk,scsus,ide,run
8	for sale	offer,sale,sell,include,drive,price,shipping,condition,system,card
9	science electronics	work,circuit,ground,power,wire,good,line,find,battery,copy
10	computer windows misc	window,file,run,driver,problem,program,work,card,system,version
11	motor motorcycle	bike,ride,dod,motorcycle,dog,good,bmw,work,rider,road
12	sport baseball	game,team,win,hit,player,run,baseball,good,play,pitch
13	religion christian	god,christian,church,jesus,christ,sin,bible,give,question,word
14	politics guns	gun,government,weapon,state,fire,law,firearm,fbi,child,day
15	computer graphics	image,file,graphic,jpeg,program,format,system,color,datum,software
16	motor autos	car,engine,drive,good,buy,problem,dealer,price,work,ford
17	religion misc	god,jesus,christian,fact,good,objective,theory,point,life,bible
18	computer windows x	window,file,program,run,server,application,widget,system,display,work
19	science crypt	key,chip,encryption,government,system,clipper,phone,security,law,information

Table 4.1: LibrAIry's 20 newsgroups topics' description [43]

In figure 4.1 we can see that our evaluation methodology was divided in two sequential stages:

1. **Modules selection** *(Section 4.1)*

   (a) ***Conversational models*** *(Section 4.1.1)*$\rightarrow$ Conversational models don't have a benchmark that can evaluate their performance for our specific task. There is no inherently wrong answer in opposition to other tasks, as translation or Question-Answering, for example. Given this fact, we decided to choose our conversational models doing a state of the art review.

   (b) ***Question templates*** *(Section 4.1.2)*$\rightarrow$ Conversational models need to receive a specifically question or order that specifies the answer we want to get. We defined some question structures to test which was was the best for our specific problem.

   (c) ***Topic words*** *(Section 4.1.3)*$\rightarrow$ Most topic models are defined by ten words that represent them, but there is no solid proof that ten is actually the number that better represents topics to discover a topic label. We decided to test which numbers of words, from one to ten, gave the best performance in our task.

   (d) ***QA models*** *(Section 4.1.4)*$\rightarrow$ Sometimes the answers that we receive from a Conversational Model not only contains the answer to our question, but other rather irrelevant information for our task. We chose to apply a Question-Answering phase where we asked the QA models to extract the topic label from the conversational models' answers. To do this, we chose some state of the art QA models and evaluated their performance in our framework.

2. **Modules evaluation** *(Section 4.2)*

   (a) ***Top words*** *(Section 4.2.1)*$\rightarrow$ We evaluated the performance of the different number of words based on the results obtained along the experiments.

   (b) ***Language models*** *(Section 4.2.2)*$\rightarrow$ We evaluated the Conversational, QA and Sentence Embedding models' performance along the experiments.

3. **System's evaluation** *(Section 4.3)*

   (a) ***Topic words relevance analysis*** *(Section 4.3.1)*$\rightarrow$ We evaluated the topic words relevance in relation to the performance of our method.

   (b) ***Topics complexity*** *(Section 4.3.2)*$\rightarrow$ We evaluated the performance of our method to produce labels for topics with different complexity.

   (c) **Topic Labelling** *(Section 4.3.3)* $\rightarrow$ During the selection evaluation we obtained a number of top words n_1 and a conversational model m_1 that worked best in the general context of our evaluation. We also got a specific combination of number of top words n_2 and conversational model m_2 that produced the best overall results for or specific task. We compared the performance obtained by these combinations, $n_1 + m_1$ and $n_2 + m_2$, with the performance of a language model fine-tuned to generate topic labels when presented with the top words n_1 and n_2.

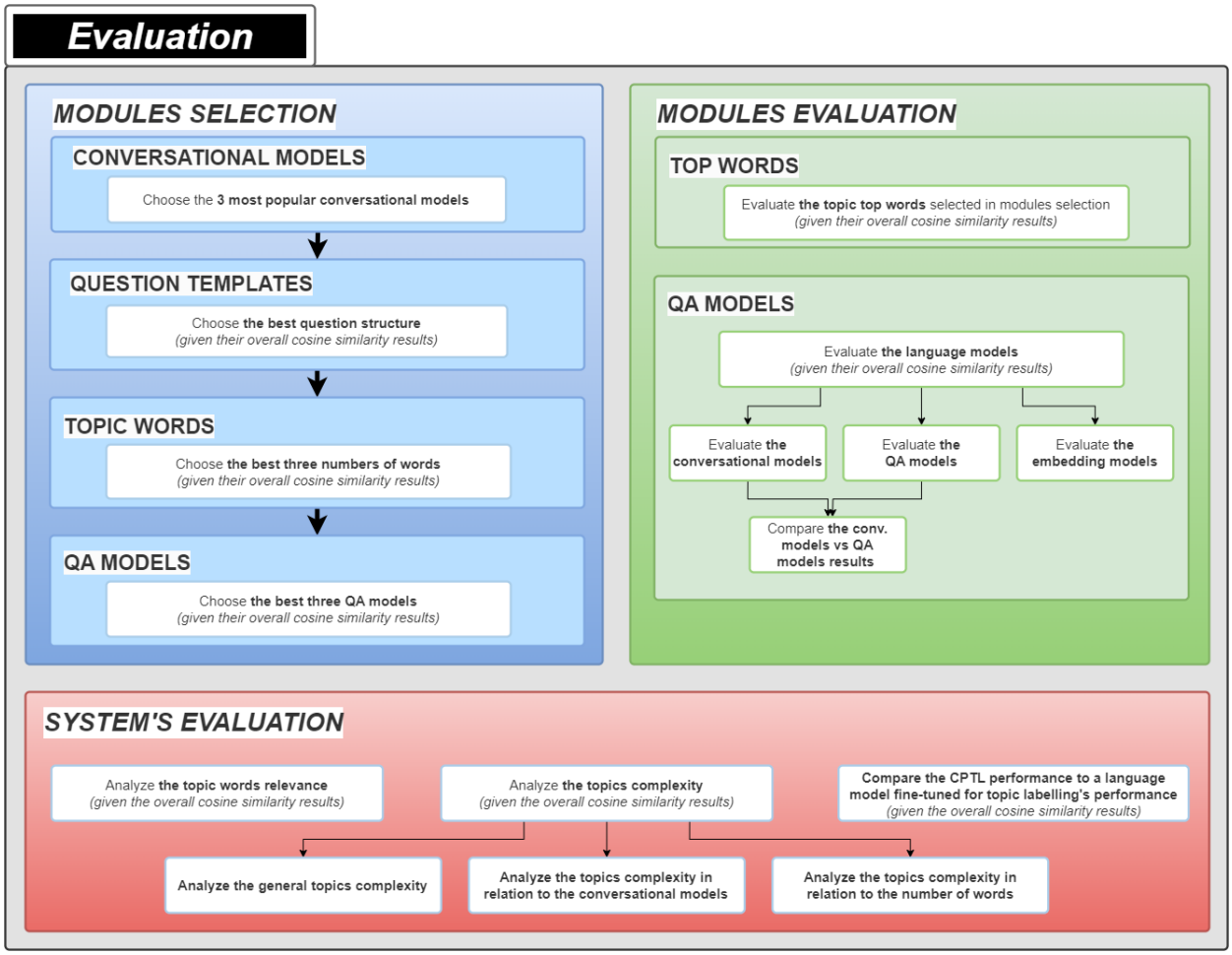


Figure 4.1: Evaluation structure

The implementation of these steps has been done using the programming language Python. The Language Models were obtained from the *HuggingFace* [44] platform and *SentenceTransformers* [45] libraries. The *code* and data in our experiments are available at: `https://doi.org/10.5281/zenodo.8018043`.

To compute the topics similarity between the real topic label and the prediction generated we needed a numerical representation that had the same shape for all the topic labels and predictions no matter the size of them. To do this, we used the *SentenceTransformers* [45] library, which allowed us to use Language Models to generate embeddings for each of them maintaining both syntactic and semantic information. *SentenceTransformers* [45] is a Python framework for state-of-the-art sentence, text and image embeddings. The initial work is described in "*Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*" [46]. Once we found a way of representing our texts, we analysed which model would potentially give us the best results. According to *SentenceTransformers*' model overview [47], shown in figure 4.2, the model with the best performance while encoding sentences is the "*all-mpnet-base-v2*" model [48], with an average performance score on encoding sentences over 14 diverse tasks of different domains of 69.57 and a speed of 2800 sentences per second on a V100GPU. The "*all-MiniLM-L6-v2*" model [49] offers an average performance score of 68.06 with a speed and size five times better than the previous one, with a speed of 14200 sentences per second on a V100GPU and a size of only 80MB opposed to the 420MB of the "*all-mpnet-base-v2*" model. We decided to use the "*all-mpnet-*

base-v2" model [48] to generate our embeddings during all the selection process given that it has a better average performance score and that we initially do not care about efficiency. In the selection evaluation, regardless, we generate the embeddings with both the "*all-mpnet-base-v2*" [48] and "*all-MiniLM-L6-v2*" [49] models to assess how much difference there is between the results given by both of them and evaluate the effect of choosing one model or the other.

Model Name	Performance Sentence Embeddings (14 Datasets)	Performance Semantic Search (6 Datasets)	Avg. Performance	Speed	Model Size
paraphrase-MiniLM-L3-v2	62.29	39.19	50.74	19000	61 MB
all-MiniLM-L6-v2	68.06	49.54	58.80	14200	80 MB
multi-qa-MiniLM-L6-cos-v1	64.33	51.83	58.08	14200	80 MB
paraphrase-multilingual-MiniLM-L12-v2	64.25	39.19	51.72	7500	420 MB
all-MiniLM-L12-v2	68.70	50.82	59.76	7500	120 MB
paraphrase-albert-small-v2	64.46	40.04	52.25	5000	43 MB
distiluse-base-multilingual-cased-v1	61.30	29.87	45.59	4000	480 MB
distiluse-base-multilingual-cased-v2	60.18	27.35	43.77	4000	480 MB
all-distilroberta-v1	68.73	50.94	59.84	4000	290 MB
multi-qa-distilbert-cos-v1	65.98	52.83	59.41	4000	250 MB
all-mpnet-base-v2	69.57	57.02	63.30	2800	420 MB
multi-qa-mpnet-base-dot-v1	66.76	57.60	62.18	2800	420 MB
paraphrase-multilingual-mpnet-base-v2	65.83	41.68	53.75	2500	970 MB

Figure 4.2: Overview of selected models of the SentenceTransformers [45] framework

4.1 Modules selection

In section 3 we presented the pipeline followed to generate the topic labels using conversational models. There we brushed over the fact that we would need four different modules to be able to generate these labels: conversational models to generate the label, question templates to ask the model for the label, the number of words that will describe the topic to label and a method to extract the label from the answers provided by the conversational models. Along the sections 4.1.1, 4.1.2, 4.1.3 and 4.1.4 we present the evaluation methodology followed to define each of these components for our specific case, that is, CPTL.

4.1.1 Conversational models

The first step of this process was selecting the conversational models to evaluate. We selected a fix number of three conversational models to evaluate, so we needed some method to assess the quality of the conversational models in order to choose the ones that are more promising. The first idea was using an already existing benchmark on conversational tasks to evaluate the possible models available. This course of action was rapidly ruled out because of the unavailability of robust benchmarks for this task. Even though there are actually several benchmark datasets designed to attempt to evaluate the performance of conversational models in their natural task, that is, having a conversation, they are not well established, as evaluating conversational models is particularly challenging due to the dynamic open-ended nature of a conversation, which can make it difficult to define clear evaluation metrics or generate representative datasets. Evaluating conversational models often requires subjective and context-dependent judgments about the quality and coherence of the generated dialogue. Furthermore, conversational models often have to handle linguistic phenomena, such as ambiguity or sarcasm, which can make it

difficult to develop standardized benchmarks that accurately reflect the complexity of natural conversations. If we specifically focus in our task, topic labelling, there aren't any benchmarks focused on evaluating this task performance specifically for conversational models as it is not an explored approach, so we can't select conversational models based in our specific task.

When thinking about how to use the different models in our Python implementation we came across *HuggingFace* [44], an open-source platform that specialized NLP and deep learning, best known for their open-source library Transformers, which provides a wide range of state-of-the-art pre-trained models for various NLP tasks, including conversational tasks. This library is built on top of deep learning frameworks such as PyTorch and TensorFlow, which makes it easy to test different models without having to modify the code each time. Hugging Face has become a popular resource for NLP researchers and developers, providing a wide range of state-of-the-art models and tools that make it easier to develop and deploy NLP applications, which makes it ideal for our purpose. HuggingFace also allows us to consult the full list of models available from their website and filter these models by task, which in our case has been "Conversational". We decided to use the data available on this website about the models to evaluate, specifically the most promising conversational models. We also based our choice on popularity and social impact, specifically for the decision of using one specific model, ChatGPT.

The results of this phase are explained in section 5.1.1.

4.1.2 Question templates

We compared three question templates or structures to generate the questions to ask the conversational models:

- Q1 $\rightarrow$ *Which is the topic that contains the words . . . ?*
- Q2 $\rightarrow$ *What is the topic related to the words . . . ?*
- Q3 $\rightarrow$ *The words . . . are related to which common topic?*

The task of topic labelling with conversational models hasn't been explored previously to our research. This task is normally approached with the exclusive use of the top words that represent the topic, so there hasn't been the need of defining a question-like structure to combine with these words. These templates were manually defined by us after pondering over how we could indicate to the conversational model the task that we wanted the model to fulfil. Conversational models are sensitive to the way a question is asked, so we needed to have different options to assess which structure is more convenient for our task. We also had to choose a sample of number of words to ask, as there were a total of 10 words describing each topic and using from 1 to 10 words for each combination of question structure and Conversational Model was too time-consuming. We decided to use each question structure with 1, 5 and 10 words for each topic and executed the following steps:

- We first generated the answer for each of the 20 topics created from the 20Newsgroup dataset for each *(1)* conversational model, *(2)* question structure and *(3)* number of words, having a total of 540 answers *(i.e. 3 conversational models $\times$ 3 question structures $\times$ 3 numbers of words $\times$ 20 topics)*. When asking the questions to one specific model, ChatGPT, we had to add a "*Please use a short answer*" at the end of each question asked. This was because this model tends to use long explanatory answers in comparison to the other models, that use a single phrase to answer. When asked "*The words 'offer' 'sale' 'sell' 'include' 'drive' are related to which common topic?*", for example, ChatGPT answers along the lines of "*These words are commonly associated with the topic of commerce or business*

transactions. "Offer" refers to presenting something for someone to consider accepting, such as a product or service. "Sale" refers to the act of exchanging goods or services for money. "Sell" refers to the act of exchanging goods or services for ...", explaining the meaning of each of the words named. We concluded that it was better to sort of "limit" the extension of these answers, so we opted to ask directly for a short answer. To the previous question, when asked for a short answer, ChatGPT answered us with a simple "*Commerce or business transactions.*". We decided not to use this final indication, "*Please use a short answer*", when asking the other models given that, from the beginning, we observed that these models tended to deliver answers that had notably less contextual information than ChatGPT and the use of the final indication "*Please use a short answer*" would only restrict even more the already brief information they provide.

- The next step was generating the embeddings for each of these answers and the topics' ground truth, that in this case would be the topics' names presented in table 4.1.

- The Cosine Similarity was computed for each pair of answer and its ground truth. Cosine similarity measures the similarity between two vectors of an inner product space, measuring the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction, as seen in figure 4.3. In our case, cosine similarity represents how similar are the predicted labels from the ground truth.

$$\text{similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Figure 4.3: Cosine Similarity formula

- Finally, to evaluate the performance, we first generated the average similarity for each combination of question structure, Conversational Model and number of words, with a total of 27 combinations. Then, with these average similarities, we rank them. With these scores, the total question score was calculated, being that each question score was the result of summing the results of the 9 combinations *(3 conversational models $\times$ 3 numbers of words)* that used each question structure.

Based on these question scores, the question structure with the top score was chosen as the best question structure and, in consequence, the question structure that was used in the next steps of the experiment.

The results of this phase are explained in section 5.1.2.

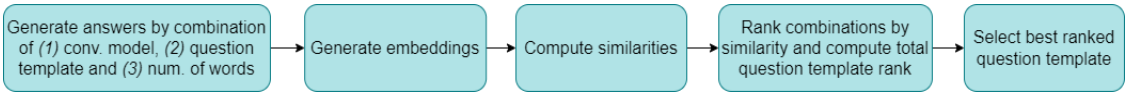


Figure 4.4: Question template selection process

4.1.3 Topic words

Once we had both the conversational models and the question structure selected, we evaluated the numbers of words, from 1 to 10 words most representative from the rank generated in the LibrAiry's model [43], as shown in table 4.1. We followed a similar process to the question structure selection process, explained in the previous section:

- We first generated the answer for each of the 20 topics for each *(1)* conversational model,*(2)* question structure and *(3)* number of words, having a total of 600 answers *(1 question structure* $\times$ *3 conversational models* $\times$ *10 numbers of words* $\times$ *20 topics*).
- The next step was generating the embeddings for each of these answers and the topics' ground truth.
- With the embeddings, the Cosine Similarity was computed for each pair of answer and its ground truth, as seen in figure 4.3.
- Finally, to evaluate the similarity results, we first generated the average similarity for each combination of question structure, Conversational Model and number of words, with a total of 30 combinations. Then, with these average similarities, we rank them. With these ranks, the total number of words rank was calculated, being that each number of words rank was the result of summing the results of the 3 combinations *(1 question structure* $\times$ *3 conversational models)* that used each number of words.

Based on these number of words' scores, the three numbers of words with the top scores were chosen and, in consequence, used in the following steps of the experiment.

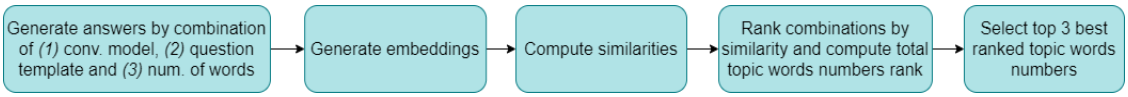


Figure 4.5: Topic words selection process

The results of this phase are explained in section 5.1.3.

4.1.4 QA models

Once we had the conversational models, the question structure and the number of words selected, the idea of using a Question-Answering Model to retrieve the labels from the answers given was raised. This is because in a lot of cases the answers given are longer than just one or two words, which, when computing the embedding, can "add" non-important information, so we decided to choose Question-Answering models to retrieve the labels from the answers and see how the results are compared to using directly the conversational models' answers.

We decided to follow an evaluation process similar to the two previous phases, but we first needed some kind of quality measure to decide which QA models were going to be evaluated. We decided to use the SQuAD 2.0 [50] benchmark to compare the QA models. The SQuAD (Stanford Question Answering Dataset) benchmark is a popular dataset in NLP that tests a model's ability to answer questions based on a given context which consists of a large dataset of Wikipedia articles, where each article is accompanied by a set of questions and corresponding answers. There has been two releases of these benchmark, SQuAD, that is focused on answerable questions based on Wikipedia articles, and SQuAD 2.0, that is an extension of SQuAD that includes unanswerable questions to test a model's ability to recognize when a question cannot

be answered from the given context. We decided to use the SQuAD 2.0's leaderboard provided by *Paperswithcode* [51]. As we can see in figure 4.6, the models with the best results based in their Exact Match are "*deberta-v3-large-squad2*" [52], "*deberta-v3-base-squad2*" [53], "*xlm-roberta-large-squad2*" [54], "*bert-large-uncased-whole-word-masking-squad2*" [55] and "*roberta-base-squad2-distilled*" [56]. We decided to use "*deepset*"'s versions of the models in the case where there are various versions of the same model, that in all cases have the same score, because of their availability on *HuggingFace* [44].

Rank	Model	Exact Match	f1	total	loss	Details	Year	Tags
1	deepset/deberta-v3-large-squad2	88.088	91.162				2022	
2	navteca/deberta-v3-base-squad2	88.088	91.162				2022	
3	deepset/deberta-v3-base-squad2	83.825	87.41				2022	
4	navteca/deberta-v3-base-squad2	83.825	87.41				2022	
5	deepset/xlm-roberta-large-squad2	81.828	84.889				2022	
6	deepset/bert-large-uncased-whole-word-masking-squad2	80.885	83.876				2022	
7	deepset/roberta-base-squad2-distilled	80.859	84.01				2022	
8	nlpconnect/roberta-base-squad2-nq	80.319	83.467				2022	
9	deepset/roberta-base-squad2	79.931	82.95	11869.0			2022	
10	weijiang2009/AlgmonQuestingAnsweringModel	79.931	82.95	11869.0			2022	

Figure 4.6: Papers with Code - SQUADv2 top 10 ranking [51]

QA models need a question and a text input to extract the answer from. In our case the question entered to the QA model was "*What is the topic that contains the words?*" in all instances and the text input was each conversational model answer. We followed a similar process to the question structure and number of words selection processes, explained in the previous sections:

- We first generated the answer for each of the 20 topics for each *(1)* conversational model, *(2)* question structure, *(3)* number of words and *(4)* QA models, having a total of 900 answers *(1 question structure* $\times$ *3 conversational models* $\times$ *3 numbers of words* $\times$ *5 QA models* $\times$ *20 topics)*.
- The next step was generating the embeddings for each of these answers and the topics' ground truth.
- With the embeddings, the Cosine Similarity was computed for each pair of answer and its ground truth, as seen in figure 4.3.
- Finally, to evaluate the similarity results, we first generated the average similarity for each combination of question structure, Conversational Model, number of words and QA Model, with a total of 45 combinations. Then, with these average similarities, we ranked them. With these ranks or scores, the total number of QA Model scores was calculated, being that each QA Model score was the result of summing the results of the 9 combinations *(1 question structure* $\times$ *3 conversational models* $\times$ *3 numbers of words)* that used each QA model.

Based on these QA models scores, the three QA models with the top 3 scores were chosen and, in consequence, the QA models that were used in the final evaluation of the experiment.

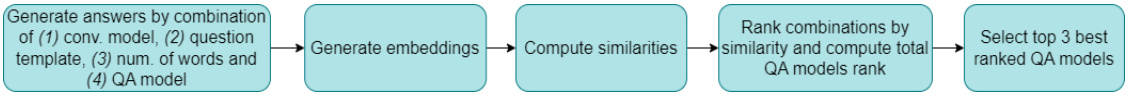


Figure 4.7: QA models selection process

The results of this phase are explained in section 5.1.4.

4.2 Modules evaluation

Once we had selected the question structure, numbers of words, Conversational and QA models, we proceeded to carry out the modules' evaluation. To do this, we used the results of all the possible combinations of the selected instances, generated following a similar process to the previous selection processes:

- We first generated the answer for each of the 20 topics for each *(1)* conversational model, *(2)* question structure, *(3)* number of words and *(4)* QA models, having a total of 180 conversational answers (1 question structure $\times$ 3 conversational models $\times$ 3 numbers of words $\times$ 20 topics) and 540 QA answers (1 question structure $\times$ 3 conversational models $\times$ 3 numbers of words $\times$ 3 QA models $\times$ 20 topics). QA models need a question and a text input to extract the answer from. In our case the question was "*What is the topic that contains the words?*" in all instances and the text input was each conversational model prediction.

- The next step was generating the embeddings for each of these answers (both conversational and QA answers) and the topics' ground truth. In this case, we generated the embeddings with "*all-mpnet-base-v2*" [48] and "*all-MiniLM-L6-v2*" [49] models, as we wanted to test out if there was a substantial difference in performance of both models.

- With the embeddings generated with both models, the Cosine Similarity was computed for each pair of answer and its ground truth, as seen in figure 4.3.

- Finally, to evaluate the similarity results, we first generated the average similarity for each combination of embedding model, question structure, Conversational Model and number of words (to evaluate the conversational models) and each combination of embedding model, question structure, Conversational Model, number of words and QA Model, with a total of 72 combinations *(2 embedding models $\times$ 1 question structure $\times$ 3 conversational models $\times$ 3 numbers of words + 2 embedding models $\times$ 1 question structure $\times$ 3 conversational models $\times$ 3 numbers of words $\times$ 3 QA models)*. The resulting similarities are divided in four main blocks: conversational models similarities with "*all-mpnet-base-v2*" [48] embeddings, conversational models similarities with "*all-MiniLM-L6-v2*" [49] embeddings, QA models similarities with "*all-mpnet-base-v2*" [48] embeddings and QA models similarities with "*all-MiniLM-L6-v2*" [49] embeddings. Then, with the average similarities, we evaluated the results by two different points of view: top words evaluation (explained in section 4.2.1) and models evaluation (explained in section 4.2.2)

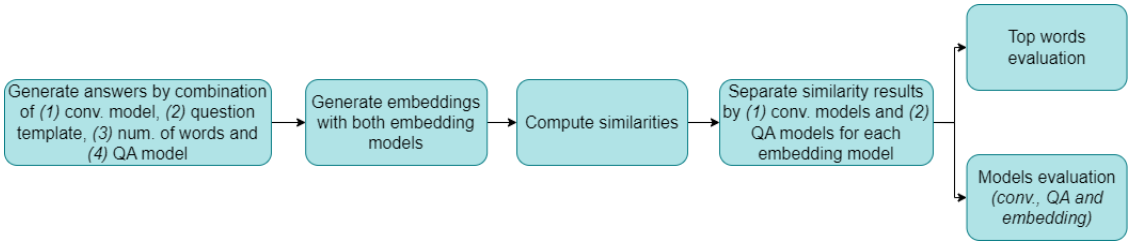


Figure 4.8: Modules evaluation process

The results of this phase are explained in section 5.3.

4.2.1 Top words

To evaluate the results by number of words, specifically the results for the top-3 number of words selected, we analysed the average ranking results by combination of Conversational or QA Model, Embedding Model and, in the case of QA models results, the QA model used. For each of these combinations we generated a ranking of the number of words, that is, each number of words had a rank from one to three, being one the best result and three the worst. With the ranking result for each number of words for each combination of Conversational or QA Model, Embedding Model and, in the case of QA models results, the QA model used, we analysed the following statistics results:

- Maximum ranking of each number of words
- Minimum ranking of each number of words
- Mean ranking
- Ranking position by mean ranking, that is, the final ranking of the top-3 number of words given by their mean ranking.

The results of this evaluation are presented in section 5.2.1.

4.2.2 Language models

To evaluate the Language Models used along these experiments, we separated the evaluation in the evaluation of the conversational models results, explained in section 4.2.2.1, the evaluation of the QA models results, explained in section 4.2.2.2, the comparison of the Conversational and QA results, explained in section 4.2.2.3, and the evaluation of the embedding models, explained in section 4.2.2.4.

4.2.2.1 Conversational models

To evaluate the conversational models' results we followed a process similar to the process explained in section 4.2.1, only analysing the results generated exclusively by the use of conversational models, without the intervention of any QA Model. We analysed the average ranking results of each Conversational Model by combination of Embedding Model and number of words. Given that we only used the top-3 conversational models, each Model had a ranking assigned from one to three, being one the top result and three the worst. With the ranking result for each conversational Model, embedding model and number of words, we analysed the following statistics results:

- Maximum ranking of each Conversational Model
- Minimum ranking of each Conversational Model
- Mean ranking
- Ranking position by mean ranking, that is, the final ranking of the top-3 conversational models given by their mean ranking.

The results of this evaluation are presented in section 5.2.2.1.

4.2.2.2 QA models

To evaluate the QA models' results we followed a process similar to the previous process analysing the results generated by the use of QA models. We analysed the average ranking results of each QA Model by Embedding Model, that is, we generated a ranking of QA model for each use of Embedding Model. Given that we only used the top-3 QA models, each Model had a ranking assigned from one to three, being one the top result and three the worst. With the ranking result for each QA and Embedding Model we analysed the following statistics results:

- Maximum ranking of each QA Model
- Minimum ranking of each QA Model
- Mean ranking
- Ranking position by mean ranking, that is, the final ranking of the top-3 QA models given by their mean ranking.

The results of this evaluation are presented in section 5.2.2.2.

4.2.2.3 Conversational vs QA models

Once we had the Conversational and QA models results, we wanted to analyse the difference in performance with and without the use of QA models to "shorten" the answers provided by the conversational models. To do this, we decided to compute the similarity difference for each Conversational Model's results, that is, QA similarity minus Conversational similarity, to see if or when, in average, the use of QA models improves the results. To do this, we analysed the following statistics results:

- Minimum difference/improvement in using QA models versus not using it.
- Maximum difference/improvement in using QA models versus not using it.
- Mean difference/improvement in using QA models versus not using it.

The results of this evaluation are presented in section 5.2.2.3.

4.2.2.4 Embedding models

To evaluate the performance of the embedding models, we compared the average similarity results given by the conversational and QA results, having a total of four average results, one for each embedding Model. We compared the difference in the similarity results given both types of embedding by conversational and QA general results. We analysed the following statistics results:

- Mean similarity by embedding Model

- Mean similarity by Language Model (conversational or QA)
- Similarity difference by Language Model (conversational or QA), that is, which is the difference in similarity using each embedding Model in both instances.
- General similarity difference between both embedding models.

The results of this evaluation are presented in section 5.2.2.4.

4.3 System's evaluation

To evaluate the overall system's performance, we defined three different evaluation processes: The analysis of the topic words relevance, in section 4.3.1, the analysis of the topics' complexity in relation to the performance presented along the evaluation, in section 4.3.2, and the evaluation of the performance of *CPTL* in comparison to a language model fine-tuned to the task of topic labelling, in section 4.3.3.

4.3.1 Topic words relevance analysis

We decided to analyse the relevance of the words that describe the topics in relation to the numbers of words. In our case, we evaluated both the specificity and the similarity of the words to the topic. The specificity was computed as the euclidean distance as shown in equation 4.9, and the cosine similarity, shown in figure 4.3.

$$\text{euclidean distance} = \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2}$$

Figure 4.9: Euclidean Distance formula between two vectors, *u* and *v*, of the same dimension, *n*

We analysed the values of the specificity and similarity for each number of words and each topic to attempt to extract information and possible patterns related to the results obtained in regard to the number of words.

The results of this evaluation are presented in section 5.3.1.

4.3.2 Topics complexity

We decided to also analyse the behaviour related to the complexity of the topics, that is, a general evaluation about the variability of the topics along the different cases (i.e. top words), explained in section 4.3.2.1, the correlation between the Conversational Models and the topics results in section 4.3.2.2, and the correlation between the number of words used and the topics results in section 4.3.2.2. We understand as complexity of the topics the complexity present in the generation of each topic's label.

4.3.2.1 General complexity

First, to analyse the variability of the topics along the different cases, that is, the different combinations of models and top words, we used the ranking of the 20 topics for each case, that is, the topics' average ranking for conversational models results using "*all-mpnet-base-v2*" [48]

embeddings, the topics' average ranking for conversational models results using "*all-MiniLM-L6-v2*" [49] embeddings, the topics' average ranking for QA models results using "*all-mpnet-base-v2*" [48] embeddings and the topics' average ranking of QA models results using "*all-MiniLM-L6-v2*" [49] embeddings. Considering these cases, for each topic we had four rankings from one to twenty, being one the topic with the best overall similarity results and twenty the worst. Given these rankings, we analysed the following statistics results:

- Maximum ranking of each topic
- Minimum ranking of each topic
- Mean ranking
- Ranking position by mean ranking, that is, the final ranking of the topics given by their mean ranking.

The results of this evaluation are presented in section 5.3.2.1.

4.3.2.2 Complexity evaluation and conversational model

To analyse the possible correlation between the Conversational Model and the topics results, we analysed the topics' results generated exclusively by the use of each Conversational Model, without the intervention of any QA Model. For each Conversational Model, we analysed the rankings of each topic, again from 1 to 20, for each embedding Model and top-3 number of words, resulting in a total of six ranking values for each topic. Given these rankings, for each conversational model we analysed the following statistics results:

- Maximum ranking of each topic
- Minimum ranking of each topic
- Mean ranking
- Ranking position by mean ranking, that is, the final ranking of the topics given by their mean ranking.

With these statistics of each conversational model and topic, we compared the results of each model and the topics' general evaluation results (explained in section 5.3.2.1) to extract possible conclusions about the correlation between the topics and the results.

The results of this evaluation are presented in section 5.3.2.2.

4.3.2.3 Complexity evaluation and number of words

To analyse the possible correlation between the number of words used and the topics results, we again analysed the topics' results generated exclusively by the use of exclusively conversational models, without the intervention of any QA Model. For each of the top-3 number of words, we analysed the rankings of each topic, again from 1 to 20, for each embedding model and conversational model, resulting in a total of six ranking values for each topic. Given these rankings, for each number of words we analysed the following statistics results:

- Maximum ranking of each topic
- Minimum ranking of each topic
- Mean ranking

- Ranking position by mean ranking, that is, the final ranking of the topics given by their mean ranking.

With these statistics of each number of words and topic, we compared the results of each number of words and the topics' general evaluation results (explained in section 5.3.2.1) to extract possible conclusions about the correlation between the topics and the results.

The results of this evaluation are presented in section 5.3.2.3.

4.3.3 Topic Labelling

In section 2.2.2 we talked about BART-TL [28], a BART-based model trained to generate topic labels given a text input. We decided to compare our results to the results generated by BART-TL. To do this, we selected two cases to be compared: the combination of the top number of words, n_1, and top conversational model, m_1, and the case with the best average similarity of all the cases tested, that has a number of words n_2 and the conversational model m_2. For the BART-TL results, we decided to use both *BART-TL-ng* and *BART-TL-all* in combination with n_1 and n_2 words.

We followed a similar process to the previous experiment methodology:

- We first generated the answer for each of the 20 topics for each model, *BART-TL-ng* and *BART-TL-all*, with n_1 and n_2 words. These models are trained to generate the label given a set of words, so in this case we only use the words as input, without the question structure.
- The next step was generating the embeddings using the best embedding model presented in section 5.2.2.4 for each of these answers and the topics' ground truth.
- With the embeddings, the Cosine Similarity was computed for each pair of answer and its ground truth, as seen in figure 4.3.
- Finally, to evaluate the similarity results, we first generated the average similarity for each combination of BART-TL model and number of words, having a total of 4 cases. We then compared the results of these four cases with the results of n_1 and n_2 words and conversational models m_1 and m_2, having a comparison between 6 total cases. To compare them, we generated the mean similarity for each case and ranked the cases by this similarity. We also further analysed the performance of each model and case for each topic to detect any anomaly or strange case.

The results of this evaluation are presented in section 5.3.3.

Chapter 5

Results and analysis

5.1 Techniques performance

Along this section we will present the performance presented by the different technical components presented in section 4.1.

5.1.1 Conversational models' selection

In section 4.1.1 we discussed a selection of conversational models based on HuggingFace's [44] top models. As seen in figure 5.1, the top three popular conversational models are "*PygmalionAI/pygmalion-6b*" [57], "*facebook/ blenderbot-400M-distill*" [58] and "*microsoft/DialoGPT-medium*" [59]. Even though "*PygmalionAI/pygmalion-6b*" [57] is the most downloaded model with a "Conversational" label, when further tested we realized that it is not a pure conversational model but a text generation model that, when entered a text in a dialogue format, given that the conversation is supposed to be between a human being and the model that we will address as "bot", generates the continuation of the dialogue not only as the "bot" but also as the human being. This is not the type of model that we aim to use in this experiment, so we decided to discard this model and keep the other two, "*facebook/blenderbot-400M-distill*" [58] and "*microsoft/DialoGPT-medium*" [59], as these two do behave as a proper Conversational Model. As for the third model, we decided to use OpenAI's *ChatGPT* [40].

ChatGPT is an AI language model that uses deep learning algorithms to generate human-like responses to natural language prompts or questions. It is a variant of the *GPT* (Generative Pre-trained Transformer) series of language models developed by OpenAI, which have been trained on large amounts of text data to learn the patterns and structures of human language. It has been specifically fine-tuned on conversational data, meaning that it has been trained to generate responses that are appropriate and relevant to the context of a conversation. *ChatGPT* has recently had a large presence on the news and society in general, as it has proven to be a very powerful tool which not only has the ability to hold a conversation, but can also carry out tasks of another nature, such as efficient analysis of datasets or code generation for health care research [1]. This social impact and its apparent potential has been the reason why we decided to use this model as our third conversational model.

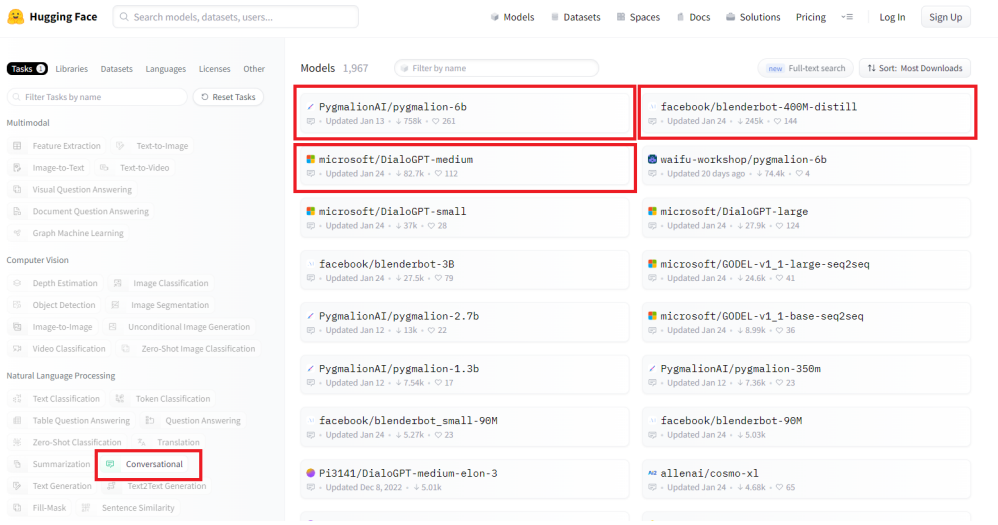


Figure 5.1: HuggingFace's most downloaded conversational models [60]

5.1.2 Question structures' composition

Once we had the conversational models selected, we define the question structure to use following the procedure explained in section 4.1.2.

- Q1 → *Which is the topic that contains the words . . . ?*
- Q2 → *What is the topic related to the words . . . ?*
- Q3 → *The words . . . are related to which common topic?*

We requested the topic label prediction for each topic for each question structure using 1, 5 and 10 words in each case for each conversational model selected, "*facebook/ blenderbot-400M-distill*" [58], "*microsoft/DialoGPT-medium*" [59] and "*ChatGPT*" [40]. We then generated the embeddings of each prediction with the "*all-mpnet-base-v2*" model [48] and computed the cosine similarity with the ground truth.

In table 5.1 we can see the evaluation results of the question structure selection *(a more complete results of the question structure selection is available in the table A.1 in Annex A)*. As we can see, the average similarity of each combination of question structure, number of words and conversational model. Given these values, we calculated the total question structure positions sum as the sum of the previous ranking generated by the average similarities and generated the ranking by question positions sum, where we concluded that the best question structure for our experiment is question 3, "*The words . . . are related to which common topic?*", given that the total question position sum is the smallest, that is, in average, the topic labels generated by the formulation of questions using this question template have a higher similarity than the other two question structures. This may be given by the fact that, in this case, we ask for a "*common*" topic, which helps the conversational model identify that the words given are related.

Question structure	Num. of words	Conversational model	Average similarity	Score	Aggregated score	Ranking by aggregated score
Q1: Which is the topic that contains the words ... ?	**1**	**blenderbot-400M-distill**	0,218965	18	129	2nd
		DialoGPT-medium	0,232794	17		
		ChatGPT	0,417945	8		
	5	**blenderbot-400M-distill**	0,238132	16		
		DialoGPT-medium	0,18795	22		
		ChatGPT	0,505575	6		
	10	**blenderbot-400M-distill**	0,157585	25		
		DialoGPT-medium	0,239013	15		
		ChatGPT	0,588928	2		
Q2: What is the topic related to the words ... ?	**1**	**blenderbot-400M-distill**	0,260996	11	146	3rd
		DialoGPT-medium	0,177007	24		
		ChatGPT	0,358362	9		
	5	**blenderbot-400M-distill**	0,211291	19		
		DialoGPT-medium	0,126813	26		
		ChatGPT	0,539788	4		
	10	**blenderbot-400M-distill**	0,181535	23		
		DialoGPT-medium	0,126286	27		
		ChatGPT	0,566372	3		
Q3: The words ... are related to which common topic?	**1**	**blenderbot-400M-distill**	0,20269	20	103	1st
		DialoGPT-medium	0,249169	13		
		ChatGPT	0,444619	7		
	5	blenderbot-400M-distill	0,196998	21		
		DialoGPT-medium	0,25149	12		
		ChatGPT	0,537663	5		
	10	blenderbot-400M-distill	0,292528	10		
		DialoGPT-medium	0,248768	14		
		ChatGPT	0,655388	1		

Table 5.1: Question structure evaluation results

5.1.3 Topic words' definition

Once we had the conversational models defined and the question structure selected, we selected the top 3 numbers of words between one and ten to use following the procedure explained in section 4.1.3. We generated the topic label prediction for each topic for each number of words from one to ten, together with each conversational model selected, "*facebook/ blenderbot-400M-distill*" [58], "*microsoft/DialoGPT-medium*" [59] and "*ChatGPT*" [40]. We again then generated the embeddings of each prediction with the "*all-mpnet-base-v2*" model [48] and computed the cosine similarity with the ground truth.

In table 5.2 we can see the evaluation results of the numbers of words selection *(a more complete evaluation results of the numbers of words selection is available in table A.2 in Annex A)*. As we can see, we generated a ranking score given the average similarity of each combination of question structure, number of words and conversational model. Given these scores, we calculated the aggregated score as the sum of the previous scores and generated the ranking by aggregated score, where we concluded that the top 3 numbers of words are, in order of score, 10, 4 and 9. If we further analyse these results both individually by model and in general, as seen in figure 5.2, we can see in dark blue, red and green the average similarity of each model for each number of words, in purple the general average of all the models and in light blue a normalized representation between 0 and 1 of the ranking of the number of words *(column "Total num. of words position sum" in table 5.2)*. In general we can see that all cases tend to peak at 4 words and then slightly decrease and increase again up until 10. When observing the graph, we can see that, even though in average 4, 9 and 10 words work better than the other numbers of words, each model has a slightly different performance at the second half of the range, from 6 to 10 words. "*facebook/ blenderbot-400M-distill*" [58] has a peak in 4, 7 and 10, in contraposition to "*microsoft/DialoGPT-medium*" [59] that peaks at 2, 4 and 9 and "*ChatGPT*" [40] that peaks at 4, 8 and 10. We can see that really the only common peak that these three models have is 4 words, which it would be a logical thought to believe that, given the previous observations and

Num. of words	Conversational model	Average similarity	Score	Aggregated score	Ranking by agregated score
1	blenderbot-400M-distill	0,20269	27	54	8th
	DialoGPT-medium	0,249169	17		
	ChatGPT	0,444619	10		
2	blenderbot-400M-distill	0,202532	28	47	6th
	DialoGPT-medium	0,296247	11		
	ChatGPT	0,553162	8		
3	blenderbot-400M-distill	0,205092	26	54	8th
	DialoGPT-medium	0,243426	21		
	ChatGPT	0,579157	7		
4	blenderbot-400M-distill	0,239985	22	39	2nd
	DialoGPT-medium	0,288027	13		
	ChatGPT	0,63414	4		
5	blenderbot-400M-distill	0,196998	29	54	8th
	DialoGPT-medium	0,25149	16		
	ChatGPT	0,537663	9		
6	blenderbot-400M-distill	0,182059	30	51	7th
	DialoGPT-medium	0,262103	15		
	ChatGPT	0,599069	6		
7	blenderbot-400M-distill	0,245272	19	45	4th
	DialoGPT-medium	0,234639	23		
	ChatGPT	0,638346	3		
8	blenderbot-400M-distill	0,231906	24	46	5th
	DialoGPT-medium	0,244167	20		
	ChatGPT	0,639141	2		
9	blenderbot-400M-distill	0,21292	25	44	3rd
	DialoGPT-medium	0,278706	14		
	ChatGPT	0,621752	5		
10	blenderbot-400M-distill	0,292528	12	31	1st
	DialoGPT-medium	0,248768	18		
	ChatGPT	0,655388	1		

Table 5.2: Topic words evaluation results

having two of the best results being 9 and 10 words, the more words, better the result, but this tendency at peaking at 4 words gives us a hint that maybe the words that describe the topics don't have the same level of relevance when generating the topic label.

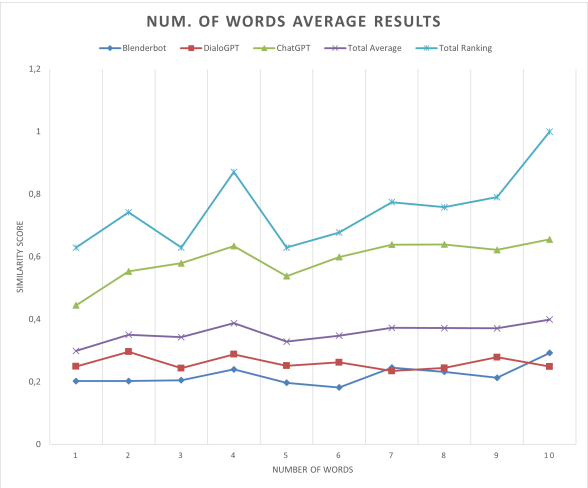


Figure 5.2: Graphic of average numbers of words results

5.1.4 QA models' selection

With the conversational models defined, the question structure and top 3 numbers of words selected, we then selected the top 3 Question-Answering models between "*deberta-v3-large-squad2*" [52], "*deberta-v3-base-squad2*" [53], "*xlm-roberta-large-squad2*" [54], "*bert-large-uncased-whole-word-masking-squad2*" [55] and "*roberta-base-squad2-distilled*" [56], following the procedure explained in section 4.1.4. Here we took the already generated predictions for 4, 9 and 10 words and each conversational model and, with each QA Model, we generated the "shortened" version of the topic label generated in each prediction. We then generated the embeddings of each QA prediction with the "*all-mpnet-base-v2*" model [48] and computed the cosine similarity with the ground truth.

QA models	Num of words	Conv. models	Average	Scores by ranking	Total QA Model Score	Ranking by QA Model score	Tie-breaker score by ranking	Tie-breaker QA Model score	Tie-breaker ranking by QA Model score
deberta-v3-large-squad2	4	blenderbot-400M-distill	0,336502	32	223	5th	-	-	-
		DialoGPT-medium	0,40795	23					
		ChatGPT	0,636448	7					
	9	blenderbot-400M-distill	0,298787	41					
		DialoGPT-medium	0,299409	40					
		ChatGPT	0,621752	13					
	10	blenderbot-400M-distill	0,413407	20					
		DialoGPT-medium	0,282605	43					
		ChatGPT	0,655388	4					
deberta-v3-base-squad2	4	blenderbot-400M-distill	0,394464	26	219	3rd	9	83	1st
		DialoGPT-medium	0,375699	28			11		
		ChatGPT	0,636448	7			3		
	9	blenderbot-400M-distill	0,347016	31			12		
		DialoGPT-medium	0,295702	42			16		
		ChatGPT	0,621752	13			6		
	10	blenderbot-400M-distill	0,405246	25			8		
		DialoGPT-medium	0,275463	44			17		
		ChatGPT	0,662094	3			1		
xlm-roberta-large-squad2	4	blenderbot-400M-distill	0,380987	27	219	3rd	10	87	2nd
		DialoGPT-medium	0,326523	33			13		
		ChatGPT	0,636448	7			3		
	9	blenderbot-400M-distill	0,326389	34			14		
		DialoGPT-medium	0,301502	39			15		
		ChatGPT	0,629236	11			5		
	10	blenderbot-400M-distill	0,41477	19			7		
		DialoGPT-medium	0,273937	45			18		
		ChatGPT	0,655388	4			2		
bert-large-uncased-whole-word-masking-squad2	4	blenderbot-400M-distill	0,43283	17	178	1st	-	-	-
		DialoGPT-medium	0,416604	18					
		ChatGPT	0,634623	10					
	9	blenderbot-400M-distill	0,364349	29					
		DialoGPT-medium	0,324137	35					
		ChatGPT	0,621722	15					
	10	blenderbot-400M-distill	0,45447	16					
		DialoGPT-medium	0,322263	36					
		ChatGPT	0,662536	2					
roberta-base-squad2-distilled	4	blenderbot-400M-distill	0,410525	22	191	2nd	-	-	-
		DialoGPT-medium	0,412481	21					
		ChatGPT	0,642029	6					
	9	blenderbot-400M-distill	0,360396	30					
		DialoGPT-medium	0,313615	38					
		ChatGPT	0,626186	12					
	10	blenderbot-400M-distill	0,406692	24					
		DialoGPT-medium	0,316729	37					
		ChatGPT	0,666603	1					

Table 5.3: QA models evaluation results

In table 5.3 we can see the reduced evaluation results of the QA models selection *(the full evaluation results of the QA models selection can be consulted in the table A.3 in Annex A)*. As we can see, "*bert-large-uncased-whole-word-masking-squad2*" [55] and "*roberta-base-squad2-distilled*" [56] are the two QA models that produced the best results, followed by "*deberta-v3-base-squad2*" [53] and "*xlm-roberta-large-squad2*" [54] tied in third place. As explained in section 4.1.4, we applied a tie-breaker between the tied models, as we needed a total of three models to execute the final evaluation. In this case, we repeated the voting method used previously, but we only took into account the results of the two tied QA models. We again assigned scores to the combinations that involved these two models according to their ranking and computed the total model score, taking the model with the highest score as the third placed QA model. This tie-breaker ended up with "*deberta-v3-base-squad2*" [53] being the third best

model, so we ended up having "*bert-large-uncased-whole-word-masking-squad2*" [55], "*roberta-base-squad2-distilled*" [56] and "*deberta-v3-base-squad2*" [53] models as top-3 QA models.

5.2 Modules performance

Once we finally had the top-3 conversational models, "*facebook/ blenderbot-400M-distill*" [58], "*microsoft/DialoGPT-medium*" [59] and "*ChatGPT*" [40], the best question structure, "*The words ... are related to which common topic?*", the top-3 numbers of words, 4, 9 and 10, and the top-3 QA models, "*bert-large-uncased-whole-word-masking-squad2*" [55], "*roberta-base-squad2-distilled*" [56] and "*deberta-v3-base-squad2*" [53], we proceeded with the modules performance evaluation. This evaluation, as explained in section 4.2, has been divided in two main blocks: top words performance evaluation (presented in section 5.2.1), and models evaluation (presented in section 5.2.2).

5.2.1 Top words

In this section we present the final number of words evaluation. As explained in section 4.2.1 and as we can see in table 5.4, given the ranking of each number of words, we generated each maximum, minimum and mean ranking, and the final ranking based on the mean ranking. This final ranking shows that the number of words that gives the best results, on general average, is 4, with a mean ranking of 1.375, followed by 10 with a mean ranking of 1.5 and 9 as last, with a mean ranking of 3. As we saw in the results presented in the selection of number of words in section 5.1.3, 4 and 10 have a relatively similar performance, which is surprising given the difference of information that, in theory, is contributed by 4 and 10 words, where it would be logical to think that 10 words would be notably more expressive than 4. This fact leads us to think that, in general, the information contributed by the first 4 words that describe the topics may be more relevant than the rest, and furthermore, that using more than 4 words can sometimes be counterproductive. We decided to analyse the relevance of the words describing the topics, in section 5.3.1, to verify if these conclusions are viable.

Number of words' statistics by ranking					
Model type	**Embedding Model**	**QA Model Name**	**Num of words**		
			4	**9**	**10**
Conv.	**all-mpnet-base-v2**	**-**	2	3	1
	all-MiniLM-L6-v2	**-**	2	3	1
QA	**all-mpnet-base-v2**	**deberta-v3-base-squad2**	1	3	2
	all-MiniLM-L6-v2	**deberta-v3-base-squad2**	1	3	1
	all-mpnet-base-v2	**bert-large-uncased-whole -word-masking-squad2**	1	3	2
	all-MiniLM-L6-v2	**bert-large-uncased-whole -word-masking-squad2**	2	3	1
	all-mpnet-base-v2	**roberta-base-squad2-distilled**	1	3	2
	all-MiniLM-L6-v2	**roberta-base-squad2-distilled**	1	3	2
Max. ranking			**1**	**3**	**1**
Min. ranking			**2**	**3**	**2**
Mean ranking			**1,375**	**3**	**1,5**
Ranking by mean ranking			**1**	**3**	**2**

Table 5.4: CPTL's top words' performance results

5.2.2 Language models

To evaluate the models used along these experiments, as explained in section 4.2.2, we separated the evaluation in the evaluation of the conversational models results, explained in section 5.2.2.1, the evaluation of the QA models results, explained in section 5.2.2.2, the comparison of the Conversational and QA results, explained in section 5.2.2.3, and the evaluation of the embedding models, explained in section 5.2.2.4.

5.2.2.1 Conversational models

In table 5.5 we can observe the ranking results for each Conversational Model. In general average, *ChatGPT* is clearly the best model, having a mean ranking of 1, while "*facebook/blenderbot-400M-distill*" and "*microsoft/DialoGPT-medium*" have a mean ranking of approximately 2.67 and 2.33 respectively. If we further analyse the similarity values presented in table 5.6, we can see that ChatGPT is widely superior to the other two models, having a mean similarity of over 0.62, while the other two models do not exceed 0.28. It is safe to say that ChatGPT proves itself again to potentially be the best conversational model publicly available *(partially)* currently and that it continues to prove itself as a powerful tool even for tasks for which it wasn't specifically trained.

conversational models statistics by ranking				
Embedding model	**Num of words**	**blenderbot-400M-distill**	**DialoGPT-medium**	**ChatGPT**
all-mpnet-base-v2	**4**	3	2	1
all-mpnet-base-v2	**9**	3	2	1
all-mpnet-base-v2	**10**	2	3	1
all-MiniLM-L6-v2	**4**	3	2	1
all-MiniLM-L6-v2	**9**	3	2	1
all-MiniLM-L6-v2	**10**	2	3	1
Max. Ranking		**2**	**2**	**1**
Min. Ranking		**3**	**3**	**1**
Mean ranking		**2,666667**	**2,333333**	**1**
Ranking by mean ranking		**3**	**2**	**1**

Table 5.5: *CPTL*'s conversational models performance results

***CPTL*'s conversational models statistics by similarity**				
Embedding model	**Num of words**	**blenderbot-400M-distill**	**DialoGPT-medium**	**ChatGPT**
all-mpnet-base-v2	**4**	0,239985	0,288027	0,63414
all-mpnet-base-v2	**9**	0,21292	0,278706	0,621752
all-mpnet-base-v2	**10**	0,292528	0,248768	0,655388
all-MiniLM-L6-v2	**4**	0,280585	0,287069	0,600634
all-MiniLM-L6-v2	**9**	0,242076	0,271729	0,60972
all-MiniLM-L6-v2	**10**	0,310256	0,261197	0,621417
Min. Similarity		**0,21292**	**0,248768**	**0,600634**
Max. Similarity		**0,310256**	**0,288027**	**0,655388**
Mean similarity		**0,263058**	**0,272583**	**0,623842**

Table 5.6: *CPTL*'s conversational models similarity results

5.2.2.2 QA models

In table 5.7 we can see the final ranking of these models, where "*bert-large-uncased-whole-word-masking-squad2*" is the best model using both embedding models. Observing the results by similarity in table 5.8, we can see that the difference between the similarity of the first ranked model and the third ranked model is only of, approximately, 0.025, all the models having average results in the range of 0.45 and 0.48.

We can't conclude that a model is better than another one out of this experiment, as these QA results are highly tied to the conversational models answers. Nevertheless, in section 5.2.2.4 we compare the Conversational and QA results to assess the performance in the use of QA models against the use of conversational models only.

QA models statistics by ranking				
QA Model		deberta-v3-base-squad2	bert-large-uncased-whole -word-masking-squad2	roberta-base-squad2-distilled
Embedding Model	all-mpnet-base-v2	3	1	2
	all-MiniLM-L6-v2	3	1	2
Max. ranking		**3**	**1**	**2**
Min. ranking		**3**	**1**	**2**
Mean ranking		**3**	**1**	**2**
Ranking by mean ranking		**3**	**1**	**2**

Table 5.7: *CPTL*'s QA models ranking results

QA models statistics by similarity			
	deberta-v3-base-squad2	bert-large-uncased-whole-word-masking-squad2	roberta-base-squad2-distilled
all-mpnet-base-v2	0,445987	0,470393	0,461695
all-MiniLM-L6-v2	0,454711	0,479283	0,469945
Min. Similarity	**0,445987**	**0,470393**	**0,461695**
Max. Similarity	**0,454711**	**0,479283**	**0,469945**
Mean similarity	**0,450349**	**0,474838**	**0,46582**

Table 5.8: *CPTL*'s QA models similarity results

5.2.2.3 Conversational vs QA models

Along these experiments we used conversational models to generate the topics' labels and QA models to shorten them, as normally labels are desired to be only made up by a few words and our ground truth labels are composed by a maximum of three words. Theoretically, the use of the QA models were supposed to improve the similarity scores as they helped to "get rid" of non-relevant information in the answer. In practice, QA models do improve the similarity scores, as seen in table 5.9, having that, for "*facebook/ blenderbot-400M-distill*" [58] and "*microsoft/DialoGPT-medium*" [59] conversational models, the answers improve a mean of 0.15 and 0.1 respectivelly. For "*ChatGPT*" [40] this improvement is quite smaller, as the answers only improve an average of 0.003, given by the fact that we already asked for short answers when asking ChatGPT to generate the topic labels *(because it tended to answer with a few lines long text)* and most answers can't be further shortened.

Even though the improvement is not huge, we can see that the use of QA models to "shorten" the answers is in fact helpful, but we have to reiterate that their performance is highly tied to the conversational model used.

Average similarity difference statistics by conversational model				
Conv. Model	**blenderbot-400M-distill**	**DialoGPT-medium**	**ChatGPT**	**General results**
Min dif.	0,133764404	0,105226768	0,000180882	0,074546112
Max dif.	0,176266449	0,140364329	0,007846037	0,110491897
Mean dif.	0,148883408	0,101277221	0,003123903	0,089936721

Table 5.9: Comparison of Conversational results and QA results

5.2.2.4 Embedding models' performance

At the beginning of the methodology section we discussed the use of two different embedding models, "*all-mpnet-base-v2*" [48] and "*all-MiniLM-L6-v2*" [49]. According to figure 4.2 "*all-mpnet-base-v2*" had a slightly better performance than "*all-MiniLM-L6-v2*", but the latter was over five times faster. Along the previous tests, we used the "*all-mpnet-base-v2*" model, as we couldn't repeat each test twice because of time and resources. We decided to actually compare the performance of these two models in the *CPTL*'s performance evaluation. In table 5.10 we can see the average similarity results for each Embedding Model for the Conversational and QA results. Curiously, in both cases "*all-MiniLM-L6-v2*" had a slightly higher similarity result than "*all-mpnet-base-v2*", but this difference, in average, was of only 0.005, so it is not a very noticeable improvement. This tells us that, in our case at least, if computation time was an important factor, as it is in a lot of cases, we could perfectly use the "*all-MiniLM-L6-v2*" model without having to sacrifice performance.

Embedding models statistics by similarity				
		Embedding Model		**Embedding avg. similarity diference**
		all-mpnet-base-v2	**all-MiniLM-L6-v2**	
Model type	**Conv.**	0,385801519	0,38718712	**0,001385602**
	QA	0,459358259	0,467979862	**0,008621603**
Mean similarity		**0,422579889**	**0,427583491**	**0,005003602**

Table 5.10: *CPTL*'s embedding models similarity results

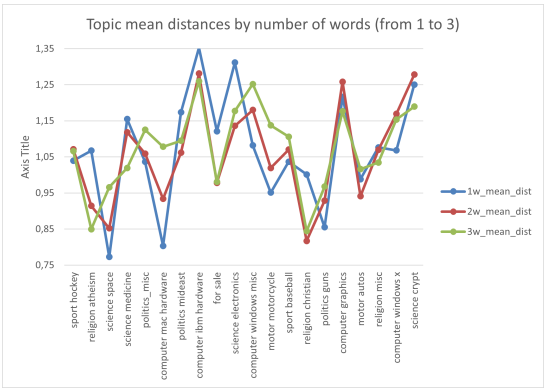
5.3 System's performance

In this section we present the overall system's performance results, separated in three main blocks: The analysis of the topic words relevance, in section 5.3.1, the analysis of the topics complexity in relation to the performance presented along the evaluation, in section 5.3.2, and the evaluation of the performance of *CPTL* in comparison to a language model fine-tuned to the task of topic labelling, in section 5.3.3.

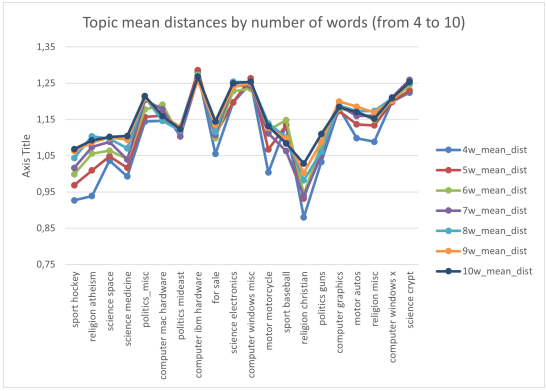
5.3.1 Topics words relevance analysis

As we stated before, we concluded that, in our case, the number of words that presented the best average results was 4, followed by 10 and 9. This was surprising because, having 9 and 10 as part of the top-3 numbers of words, a logical thought would be that the more words we use, the better results we get, but our experiments reflect that 4 words has a slightly better performance than the others. This fact leads us to the evaluation of the balance of relevance of the words that describe the topics and its possible relation with our results. To evaluate this relevance, as explained in section 4.3.1, we generated the average distance, computed with the euclidean distance, and cosine similarity of 1 to 10 words in relation to the real topic label.

In figures 5.3 and 5.4 we can observe the mean distances and similarities by number of words from 1 to 10 words, specifically from 1 to 3 words in subfigures *a)* and from 4 to 10 words in subfigures *b)*. By observing the mean distances and similarities by number of words in both cases, that is by showing the progress graphically for each topic, it can be intuited that during the first three number of words, from 1 to 3, shown in subfigures *a)* in both cases, there is less stability of the distances and similarities and, from 4 words onwards, shown in subfigures *b)*, some kind of pattern is defined. We can also see that, by looking at the mean distances and similarities of 4 to 10 words, 4 words generally have the shortest distance on most topics and the highest similarity. This may be the reason why 4 words performs so well, because, while it is true that with 1 to 3 words there are better distances and similarities than 4, we speculate that the linguistic model does not get enough information with so few terms, given the fact that the stability in the predictions starts to appear at 4 words. This, together with the fact that from 4 to 10 words, 4 words generally have the shortest distance on most topics and the highest similarity, may be the reason why 4 words is, in average, 4 words present the best results.

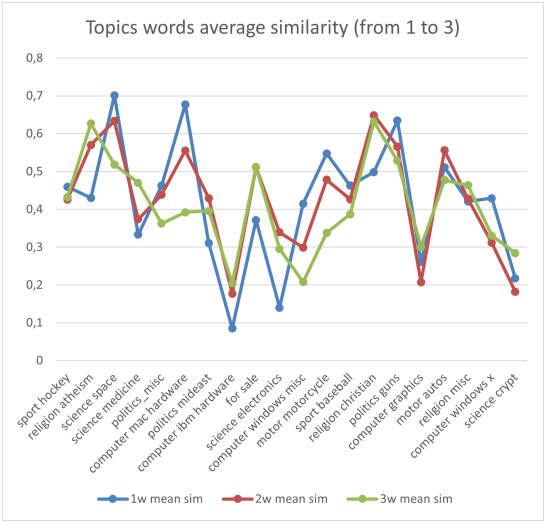


(a)

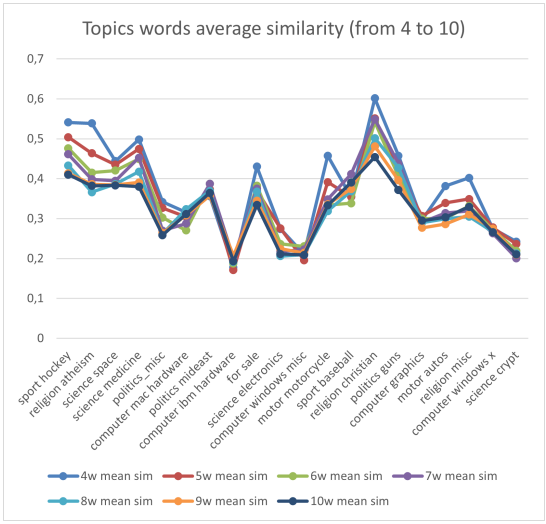


(b)

Figure 5.3: Topics words mean distance to the topic by number of words



(a)



(b)

Figure 5.4: Topics words average similarity to the topic by number of words

5.3.2 Topics' complexity

As explained along section 4.3.2, we decided to analyse the performance results from a topic point of view, that is, a general evaluation about the ranking variability of the topics along the different cases tested, which results are presented in section 5.3.2.1, the correlation between the conversational models and the results of each topic in section 5.3.2.2, and finally the correlation between the number of words used to generate the label and the topics results in section 5.3.2.3.

5.3.2.1 Topics' general complexity

Given the general ranking of each topic for the conversational and QA results using "*all-mpnet-base-v2*" [48] and "*all-MiniLM-L6-v2*" [49] models, we generated a series of statistics available in table 5.11, where we can see the final ranking of topics based on their mean ranking. This ranking represents the average results of the topics labels, being 1 the best labelled model in average, and 20 the worst. As we can see, "*sports hockey*" is, without doubt, the topic best labelled in average along the cases. On the other hand, "*science crypt*" is the worst labelled topic. If we observe figure 5.5, we can analyse the ranking variability of the different topics in general average. There are topics that are very stable in the ranking, as "*sport hockey*" and "*computer graphics*", opposed to others that have been ranked in very diverse positions, as "*politics mideast*", that has a minimum ranking of 20 and a maximum of 10.

Topics' general statistics										
Answer Model Type			Conv.		QA.		Min. ranking	Max. ranking	Mean ranking	Final ranking based on mean ranking
Embedding Model			all-mpnet -base-v2	all-MiniLM -L6-v2	all-mpnet -base-v2	all-MiniLM -L6-v2				
Topics	0	sport hockey	1	1	1	1	**1**	**1**	**1**	**1**
	1	religion atheism	4	5	2	4	**2**	**5**	**3,75**	**4**
	2	science space	7	7	5	5	**5**	**7**	**6**	**6**
	3	science medicine	13	8	13	11	**8**	**13**	**11,25**	**11**
	4	politics misc	16	15	15	16	**15**	**16**	**15,5**	**15**
	5	computer mac hardware	3	2	4	2	**2**	**4**	**2,75**	**2**
	6	politics mideast	10	20	12	19	**10**	**20**	**15,25**	**14**
	7	computer ibm hardware	15	18	19	20	**15**	**20**	**18**	**19**
	8	for sale	19	14	18	12	**12**	**19**	**15,75**	**16**
	9	science electronics	12	9	6	7	**6**	**12**	**8,5**	**8**
	10	computer windows misc	14	19	14	18	**14**	**19**	**16,25**	**17**
	11	motor motorcycle	6	3	8	6	**3**	**8**	**5,75**	**5**
	12	sport baseball	5	6	9	8	**5**	**9**	**7**	**7**
	13	religion christian	2	4	3	3	**2**	**4**	**3**	**3**
	14	politics guns	8	11	7	9	**7**	**11**	**8,75**	**9**
	15	computer graphics	9	10	10	10	**9**	**10**	**9,75**	**10**
	16	motor autos	17	12	16	13	**12**	**17**	**14,5**	**13**
	17	religion misc	11	13	11	14	**11**	**14**	**12,25**	**12**
	18	computer windows x	18	16	17	15	**15**	**18**	**16,5**	**18**
	19	science crypt	20	17	20	17	**17**	**20**	**18,5**	**20**

Table 5.11: Topics general complexity results

These results on their own do not notably contribute to the understanding of the results, so we decided to analyse if there was a correlation between the topic results exposed previously and the election of Conversational Model, in section 5.3.2.2, and/or number of words, in section 5.3.2.3.

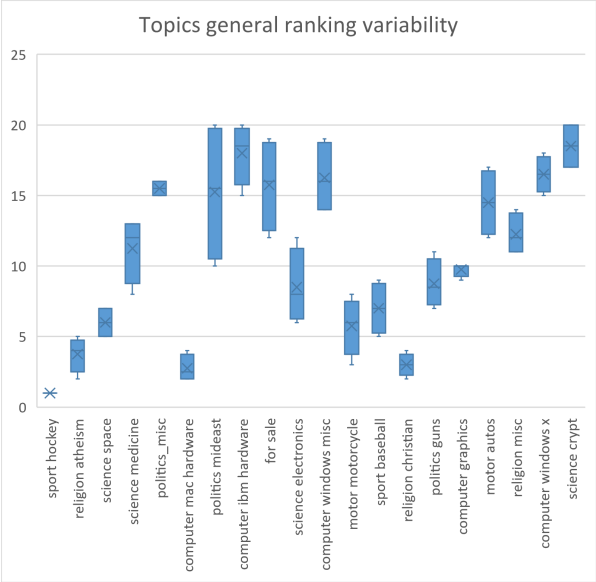
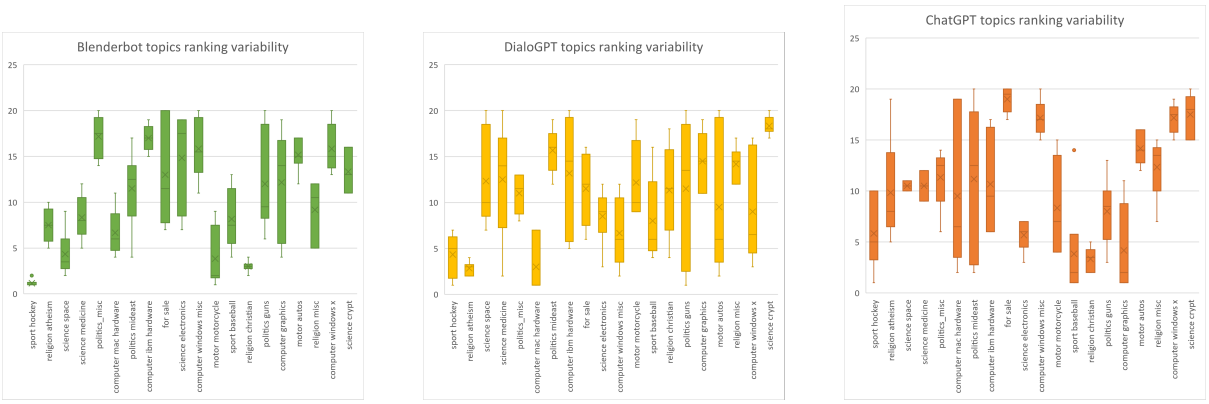


Figure 5.5: Topics general ranking variability

5.3.2.2 Based on conversational models

In this instance, given the general ranking of each topic for the conversational results using "*all-mpnet-base-v2*" [48] and "*all-MiniLM-L6-v2*" [49] models, we generated a series of statistics available in table 5.12, where we can see the final ranking of topics based on their mean ranking for each conversational model. If we observe figure 5.6 where the topics ranking variability is shown for each conversational model, we can see that the variability of the topics is different for each case. If we observe "sport hockey", for example, we can see that for *blenderbot-400M-distill* [58] we have a very similar variability to the one it had in the general evaluation presented previously, having a mean ranking of 1.17 in comparison to the 1.0 it had before, but for the other two conversational models it has a mean ranking of 4.34 and 5.83, respectively.

Given the facts presented before, we could conclude that the conversational model selected to generate the labels is relevant to the results of the topics obtained, as we've already seen that the variability of each topic is different for each model and we see no common pattern on the results obtained with each Model.



(a) Blenderbot (b) DialoGPT (c) ChatGPT

Figure 5.6: Topics ranking variability by Conversational Model

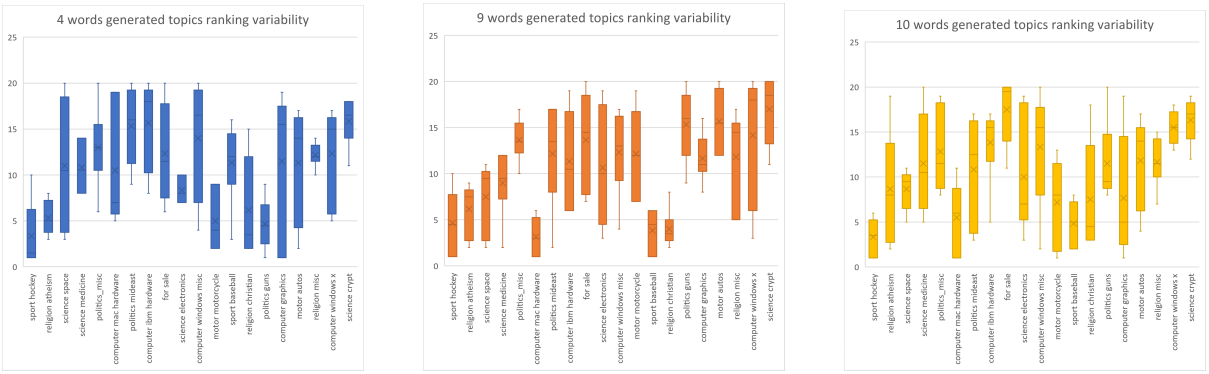
Topics' statistics by Conversational Model rankings														
Conv. Model			blenderbot-400M-distill				DialoGPT-medium				ChatGPT			
Statistic			Min. ranking	Max. ranking	Mean ranking	Final ranking based on mean ranking	Min. ranking	Max. ranking	Mean ranking	Final ranking based on mean ranking	Min. ranking	Max. ranking	Mean ranking	Final ranking based on mean ranking
Topics	0	sport hockey	1	2	1,166667	1	1	7	4,333333	3	1	10	5,833333	5
	1	religion atheism	5	10	7,5	6	2	4	2,833333	1	5	19	9,833333	9
	2	science space	2	9	4,333333	4	7	20	12,33333	14	10	11	10,5	10
	3	science medicine	5	12	8,333333	8	2	20	12,5	15	9	12	10,5	10
	4	politics_misc	14	20	17,16667	20	8	13	11	9	6	14	11,33333	14
	5	computer mac hardware	4	11	6,666667	5	1	7	3	2	2	19	9,5	8
	6	politics mideast	4	17	11,5	10	12	19	15,66667	19	2	20	11,16667	13
	7	computer ibm hardware	15	19	17	19	5	20	13,16667	16	6	17	10,66667	12
	8	for sale	7	20	13	13	6	16	11,5	11	17	20	19	20
	9	science electronics	7	19	14,83333	15	3	12	8,5	6	3	7	5,666667	4
	10	computer windows misc	11	20	15,83333	17	2	12	6,666667	4	15	20	17,16667	17
	11	motor motorcycle	1	9	3,833333	3	9	19	12,16667	13	4	15	8,333333	7
	12	sport baseball	4	13	8,166667	7	4	16	8	5	1	14	3,833333	2
	13	religion christian	2	4	3	2	4	18	11,33333	10	2	5	3,333333	1
	14	politics guns	6	20	12	11	1	20	11,5	11	3	13	8	6
	15	computer graphics	4	19	12,16667	12	11	19	14,5	18	1	11	4,166667	3
	16	motor autos	12	17	15,16667	16	2	20	9,5	8	12	16	14,16667	16
	17	religion misc	5	12	9,166667	9	12	17	14,16667	17	7	15	12,33333	15
	18	computer windows x	13	20	15,83333	17	3	17	9	7	15	19	17,16667	17
	19	science crypt	11	16	13,33333	14	17	20	18,33333	20	15	20	17,5	19

Table 5.12: Topics' statistics by conversational model rankings

5.3.2.3 Based on number of words

Given the general ranking of each topic for the conversational results using "*all-mpnet-base-v2*" [48] and "*all-MiniLM-L6-v2*" [49] models, we generated a series of statistics available in table 5.13, where we can see the final ranking of topics based on their mean ranking for each number of words 4, 9 and 10. If we observe figure 5.7 where the topics ranking variability is shown for each number of words *(4, 9 and 10)*, we can see a similar case as the previous one, having that the variability of the topics is again different for each case. It is true that, in this case, the variability of each topic does not differ as much as the previous one, but we also can't conclude that there is a pattern along the three cases, as there are a few topics that do notably differ between cases, as the topic "*religion atheism*", where the variability increases with the increase in number of words.

Similar to the previous section, we can conclude that the number of words used is relevant to the results of the topics obtained, as we've already seen that the variability of each topic is different for each number of words and we see no common pattern on the results obtained with each Model. We also observe that the use of one number of words over the others does not reduce the variability of the topics results.



(a) 4 words (b) 9 words (c) 10 words

Figure 5.7: Topics ranking variability by number of words

Topics' statistics by number of words' rankings														
Num. of words			4				9				10			
Statistic			Min. ranking	Max. ranking	Mean ranking	Final ranking based on mean ranking	Min. ranking	Max. ranking	Mean ranking	Final ranking based on mean ranking	Min. ranking	Max. ranking	Mean ranking	Final ranking based on mean ranking
Topics	0	sport hockey	1	10	3,333333	1	1	10	4,666667	4	1	6	3,333333	1
	1	religion atheism	3	8	5,333333	4	2	9	6,166667	5	2	19	8,666667	7
	2	science space	3	20	11	9	2	11	7,5	6	5	11	8,666667	7
	3	science medicine	8	14	10,83333	8	2	12	9	7	5	20	11,5	11
	4	politics_misc	6	20	13	16	10	17	13,66667	15	8	19	12,83333	15
	5	computer mac hardware	5	19	10,5	7	1	6	3,166667	1	1	11	5,5	3
	6	politics mideast	9	20	15,33333	18	2	17	12,16667	12	3	17	10,83333	10
	7	computer ibm hardware	8	20	15,66667	19	6	19	11,33333	9	5	17	13,83333	17
	8	for sale	6	20	12,33333	14	7	20	13,66667	15	11	20	17,5	20
	9	science electronics	7	10	8,333333	6	3	19	10,66667	8	3	19	10	9
	10	computer windows misc	4	20	14	17	4	17	12,33333	14	2	20	13,33333	16
	11	motor motorcycle	2	9	5	3	7	19	12,16667	12	1	13	7,166667	4
	12	sport baseball	3	16	11,33333	10	1	6	3,833333	2	2	8	4,833333	2
	13	religion christian	2	15	6,166667	5	2	8	4	3	3	18	7,5	5
	14	politics guns	1	9	4,666667	2	9	20	15,33333	18	8	20	11,5	11
	15	computer graphics	1	19	11,5	12	8	16	11,66667	10	1	19	7,666667	6
	16	motor autos	2	17	11,33333	10	12	20	15,66667	19	4	17	11,83333	14
	17	religion misc	10	14	12,16667	13	5	17	11,83333	11	7	15	11,66667	13
	18	computer windows x	5	17	12,33333	14	3	20	14,16667	17	13	18	15,5	18
	19	science crypt	11	18	15,83333	20	11	20	17	20	12	19	16,33333	19

Table 5.13: Topics' statistics by number of words' rankings

5.3.3 Topic Labelling

As explained in the evaluation section, we generated the results of *BART-TL-ng* and *BART-TL-all* [28] for n_1 and n_2 words of each topic and compared them to the results of the combination of the top number of words, n_1, and top conversational model, m_1, and the case with the best average similarity of all the cases tested, that has a number of words n_2 and the conversational model m_2, where in this case m_1 and m_2 are both ChatGPT (without QA Model) and n_1 and n_2 are 4 and 10 words respectively, using the embeddings generated with the "*all-MiniLM-L6-v2*" model, as shown in table 5.14. Here we can see the similarity of each topic for each case together with the mean similarity of each case. We can see that the obvious winner is ChatGPT with both 4 and 10 words, as they have a mean similarity of over 0.6 while the other cases do not surpass the 0.4 mark. We can also observe that *BART-TL-ng* and *BART-TL-all* deliver similar results given the same number of words, having that for 4 words they have a similarity of 0.39 and 0.38 and for 10 words they have a similarity of 0.3 and 0.31. It is intriguing the fact that BART-TL has much better similarity using only 4 words than using 10 words, with a difference in similarity of approximately 0.1, contrary to ChatGPT, that has a similarity result of 0.66 using 10 words and 0.60 using 4 words.

If we analyse the behaviour of the models in a deeper way we can see that the performance of Bart-TL-ng on 4 and 10 words coincides in the topics 2, 5, 6, 7, 8, 13 and 17. This is because the model gives the same answer in both models. Knowing that BART-TL-ng is trained for "large-scale text generation tasks", it is logical to think that it acts in a similar way with 4 and 10 words since that cannot be called a "large" text. It is curious to see that many of the tags are repeated, at first we thought that it acted a bit like ChatGPT, which seems to look at the relationship between words, but in this case it seems that it has a tendency to use specific tags. If we look closely, there is a specific label that is repeated several times without apparent meaning "on the other hand", which leads us to think that perhaps the model presents a slight "bias" towards some specific tags, especially the Bart-TL-all model.

On topic 11, "*motor motocycle*", the performance of both Bart-TL models are drastically better on 4 words than it is on 10 words. We know that the words that describe the 11th topic are "bike, ride, dod, motorcycle, dog, good, bmw, work, rider, road". If only the 4 words are taken, 3 of the 4 refer to bicycles or motorcycles, but if all the words are taken into account, 4 of them,

"dod, dog, good, work" have no apparent meaning within the topic, which can make it more difficult to extrapolate the common topic. This may be the reason why the models perform better with only 4 words than with 10 words and why, in general, the 4 words results are better.

Topics 12 and 13 generate the same performance in three Bart-TL variants, except for the one that considers 10 words of Bart-TL-ng and 10 words of Bart-TL-all, respectively. Here, the way of fine-tuning both Bart-TL models possibly intervenes together with the "bias" factor explained before. Bart-TL-ng was trained with data they call "base dataset" *(based on NETL labeler)* and a set of n-grams generated from the most important words of the topics, while Bart-TL-all was fine-tuned, apart from the data used in Bart-TL-ng, with groups of sentences and common noun phrases from the corpus. In both topics 12 and 13, three of the Bart-TL cases give the same answer while one of the tests gives another, but looking at what labels they predict they actually are not far-fetched. In topic 12 "sport baseball" three of the tests predict "the game", while the other one predicts "result", and topic 13 "religion christian" three of the tests predict "christianity", while the other is "baptism". In both cases it can be seen related to the topic of the prediction, therefore we suppose that the difference can come from what has been learned in the fine-tuning. In topic 14, "politics guns" we have the same case of three answers being the same, but here we have that three of the tests predict "on the other hand" as label, while Bart-TL-ng with 10 words predicts "Federal law". Knowing that this topic is described by the words "gun, government, weapon, state, fire, law, firearm, fbi, child, day" and knowing that Bart-TL-ng has been trained with n-grams, we wonder if it might recognize the "n-grams" or syllables of the words "gun", "state", "law"... which are concentrated after the 4th word and, again, arrive to the assumption that the difference comes from what the models have learned during the fine-tuning, having the possibility that Bart-TL-all has been repeatedly trained with sentences containing the construction "on the other hand".

There are several topics where the Bart-TL models improve the performance offered by ChatGPT with 10 words, specifically the topics 8, 10, 11, 16. In this case, added to everything exposed previously, we noticed that ChatGPT tends to be a bit more general in terms of topics. For example when referring to Windows it calls it "Operating System", as opposed to Bart-TL which tends to be more specific, where in the same case it predicts "Windows 8". Although it is true that ChatGPT with 4 words tends to have the same or better similarity in most of these topics, except in topic 10 where it drops slightly, in three of these topics Bart-TL improves the similarity slightly because its answers are more specific, although the most notable case is the case of topic 11, "motor motorcycle", where Bart-TL-ng with 4 words predicts the label "motorcycle", but keep in mind that in this case "motorcycle" is one of the topic description words, specifically the fourth.

Finally, looking at the averages of each model, it is clear that Bart-TL is more sensitive to topic words than ChatGPT. This might happen because ChatGPT seems to analyse the set of words as a set and taking into account their semantic sense, while Bart-TL seems to take the words more separately and syntactically, therefore hindering the detection of relationships between words in a more general way. Also, as explained before, Bart-TL seems to have a slight "bias" towards specific words or labels in various cases, giving us the hint that there may be minimal over-fitting on the data used in fine-tuning.

models statistics by similarity								
Num. words			4	10	4	10	4	10
Model			ChatGPT	ChatGPT	bart-tl-ng	bart-tl-ng	bart-tl-all	bart-tl-all
Topics	0	sport hockey	0,74445	0,82280	0,45501	0,21021	0,45501	0,05375
	1	religion atheism	0,64873	0,37268	0,19122	0,13700	0,58622	0,53602
	2	science space	0,64219	0,68239	0,35000	0,35000	0,35000	0,35000
	3	science medicine	0,64464	0,66918	0,19096	0,27939	0,25207	0,16808
	4	politics_misc	0,57206	0,69698	0,22657	0,18500	0,22657	0,33146
	5	computer mac hardware	0,36812	0,75000	0,38720	0,38720	0,43368	0,38720
	6	politics mideast	0,31487	0,87233	0,11392	0,11392	0,26156	0,36294
	7	computer ibm hardware	0,60984	0,49537	0,42399	0,42399	0,43643	0,32937
	8	for sale	0,45003	0,28843	0,40494	0,40494	0,16241	0,40494
	9	science electronics	0,67017	0,79020	0,31248	0,32354	0,05541	0,16245
	10	computer windows misc	0,44330	0,46579	0,48528	0,21494	0,34219	0,21494
	11	motor motorcycle	0,74450	0,66651	0,90417	0,07602	0,90417	0,07602
	12	sport baseball	0,75195	0,87238	0,45900	0,18796	0,45900	0,45900
	13	religion christian	0,76317	0,83095	0,82934	0,82934	0,82934	0,47433
	14	politics guns	0,70535	0,69824	0,18653	0,33396	0,18653	0,18653
	15	computer graphics	0,88306	0,87297	0,22740	0,23679	0,22740	0,57956
	16	motor autos	0,56186	0,56803	0,57933	0,19520	0,08862	0,19520
	17	religion misc	0,52233	0,75335	0,63954	0,63954	0,63954	0,63954
	18	computer windows x	0,46082	0,55562	0,38353	0,18191	0,38353	0,28665
	19	science crypt	0,51123	0,38355	0,14597	0,19291	0,36119	0,02164
Mean similarity			**0,600634**	**0,655388**	**0,394819**	**0,295188**	**0,382044**	**0,310982**
Ranking by mean similarity			**2**	**1**	**3**	**6**	**4**	**5**

Table 5.14: BART-TL results comparison

Chapter 6

Conclusions and future work

Throughout this document, we have presented our evaluation of conversational models' ability to generate labels for topics based on a given set of representative words. We established a systematic approach for sequentially selecting a subset of the most remarkable publicly accessible models to carry out a constrained study. We also compared the performance of these models with a task-specific model, *BART-TL* [28], to assess the differences in using both types of models.

We know that, in general, conversational models might have a wider range of utilization beyond a conversational use. Along our experiments we have seen that these models do present the potential to be used to generate topic labels, as *ChatGPT* achieved a similarity notably better than a task-specific model, but if we take a look at the bigger picture, taking into account the other two conversational models used, this might not be something that can be yet done in a general sense. To execute these experiments we used three different conversational models, two models available in *HuggingFace* [44] that are of free access and anyone can manipulate, and *ChatGPT*, that for the time being can only be used accessing their website based interface and has a restricted use. If we compare the results of the two publicly available models and *ChatGPT*, we can see that there is a huge difference in performance, having that *ChatGPT* presents results over twice as good as the other two. This is possibly because *ChatGPT* is a conversational model that has *OpenAI's GPT* models behind, that required of thousands of GPUs, hundreds of gigabytes of data and weeks or months to train. At the time of writing this document, *ChatGPT* was working over *GPT-3* specifically, which requires several hundreds of gigabytes of storage space given its 175 billion parameters. This is obviously not something that can be achieved by a common user, as the economical and computational power that is required to train and run this kind of models is out of reach for the vast majority of people and only big corporations, such as Microsoft in this case, can have. Given these facts, we can't say that conversational models can be used to generate topic labels currently, but the prospecting is that, in the future, we might not need to train models specifically to generate this labels, but we might be able to do it with general purpose models.

On the other hand, along the experiments, we worked with a variety of numbers of words to represent the topics when asking for the topic labels. These experiments have brought to light the relevance of the topic modelling method employed to extract the description words of the topics and the no linearity relevance of these words related to the topic representation. A lot of times, when working with a group of topics, they are all represented with an equal fix number of words, but we have seen that not all topics are better represented with a static number of words, but rather, as seen in our topic swords relevance analysis, even though there is a clear benefit of the general use of 4 words, each topic is better represented with a specific individual number

of words, which in some cases might be 4, but in others might be 5 or 10, for example. With this we can answer ***RQ4***, "*What is the most adequate number of words to describe a probabilistic topic?*", and say that sometimes less is more, and that we don't really need to use a big amount of words to represent our topics.

If we observe table 6.1, then, we can observe that we were able to answer all our research questions: We defined a pipeline to generate probabilistic topic labels *(RQ1)* and explained how to extract the topic labels from the conversational models *(RQ2)* in section 3. We also evaluated the performance of our proposed method *(RQ3)* and analysed the most adequate number of words to describe a probabilistic topic *(RQ4)* in sections 5 and 6.

ID	Research question	Answer
RQ1	How can conversational models be used to generate labels for probabilistic topics?	1. Top words selection 2. Question composition 3. Question formulation 4. Label extraction
RQ2	How can topic labels be extracted when given a conversational model's answer?	Using QA models
RQ3	What is the quality of the topic labels generated by the conversational models?	Max. similarity of 0.655
RQ4	What is the most adequate number of words to describe a probabilistic topic?	It depends of the topic represented

Table 6.1: Research questions' answers

Regarding the future work, we would like to further test the capacity that conversational models have to create topic labels using different conversational models to the ones used and a wider variety of topics, using for example *LibrAIry*'s DBpedia topics dataset [43] which contains 405 topic models. We also have previously stated that the number of words to represent the topics might be different in each case, so it might be interesting to study an approach to topic modelling that does not use a fix number of words when modelling the topics, but rather consider the relevance of the words in regard to the topic represented.

Bibliography

[1] M. Sallam, "Chatgpt utility in healthcare education, research, and practice: Systematic review on the promising perspectives and valid concerns," *Healthcare*, vol. 11, no. 6, 2023, ISSN: 2227-9032. DOI: 10.3390/healthcare11060887. [Online]. Available: https://www.mdpi.com/2227-9032/11/6/887.

[2] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, no. 4, pp. 77–84, Apr. 2012, ISSN: 0001-0782. DOI: 10.1145/2133806.2133826. [Online]. Available: https://doi.org/10.1145/2133806.2133826.

[3] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 50–57.

[4] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," in *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14, MIT Press, 2001. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2001/file/296472c9542ad4d4788d543508116cbc-Paper.pdf.

[5] T. Griffiths, M. Jordan, J. Tenenbaum, and D. Blei, "Hierarchical topic models and the nested chinese restaurant process," in *Advances in Neural Information Processing Systems*, S. Thrun, L. Saul, and B. Schölkopf, Eds., vol. 16, MIT Press, 2003. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2003/file/7b41bfa5085806dfa24b8c9de0ce567f-Paper.pdf.

[6] D. J. Aldous, "Exchangeability and related topics," in *École d'Été de Probabilités de Saint-Flour XIII — 1983*, P. L. Hennequin, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 1–198, ISBN: 978-3-540-39316-0.

[7] J. Lafferty and D. Blei, "Correlated topic models," in *Advances in Neural Information Processing Systems*, Y. Weiss, B. Schölkopf, and J. Platt, Eds., vol. 18, MIT Press, 2005. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2005/file/9e82757e9a1c12cb710ad680db11f6f1-Paper.pdf.

[8] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06, Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 113–120, ISBN: 1595933832. DOI: 10.1145/1143844.1143859. [Online]. Available: https://doi.org/10.1145/1143844.1143859.

[9] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the 2009 conference on empirical methods in natural language processing*, 2009, pp. 248–256.

[10] X. Cheng, X. Yan, Y. Lan, and J. Guo, "Btm: Topic modeling over short texts," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 2928–2941, 2014. DOI: 10.1109/TKDE.2014.2313872.

[11] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, ser. SIGIR '03, Toronto, Canada: Association for Computing Machinery, 2003, pp. 267–273, ISBN: 1581136463. DOI: 10.1145/860435.860485. [Online]. Available: https://doi.org/10.1145/860435.860485.

[12] T. Shi, K. Kang, J. Choo, and C. K. Reddy, "Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations," in *Proceedings of the 2018 World Wide Web Conference*, ser. WWW '18, Lyon, France: International World Wide Web Conferences Steering Committee, 2018, pp. 1105–1114, ISBN: 9781450356398. DOI: 10.1145/3178876.3186009. [Online]. Available: https://doi.org/10.1145/3178876.3186009.

[13] Q. Mei, C. Liu, H. Su, and C. Zhai, "A probabilistic approach to spatiotemporal theme pattern mining on weblogs," in *Proceedings of the 15th International Conference on World Wide Web*, ser. WWW '06, Edinburgh, Scotland: Association for Computing Machinery, 2006, pp. 533–542, ISBN: 1595933239. DOI: 10.1145/1135777.1135857. [Online]. Available: https://doi.org/10.1145/1135777.1135857.

[14] T. Atapattu and K. Falkner, "A framework for topic generation and labeling from mooc discussions," in *Proceedings of the Third (2016) ACM Conference on Learning @ Scale*, ser. L@S '16, Edinburgh, Scotland, UK: Association for Computing Machinery, 2016, pp. 201–204, ISBN: 9781450337267. DOI: 10.1145/2876034.2893414. [Online]. Available: https://doi.org/10.1145/2876034.2893414.

[15] X. Wang, A. McCallum, and X. Wei, "Topical n-grams: Phrase and topic discovery, with an application to information retrieval," in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, 2007, pp. 697–702. DOI: 10.1109/ICDM.2007.86.

[16] Q. Mei, X. Shen, and C. Zhai, "Automatic labeling of multinomial topic models," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '07, San Jose, California, USA: Association for Computing Machinery, 2007, pp. 490–499, ISBN: 9781595936097. DOI: 10.1145/1281192.1281246. [Online]. Available: https://doi.org/10.1145/1281192.1281246.

[17] J. H. Lau, D. Newman, S. Karimi, and T. Baldwin, "Best topic word selection for topic labelling," in *Coling 2010: Posters*, 2010, pp. 605–613.

[18] A. E. Cano Basave, Y. He, and R. Xu, "Automatic labelling of topic models learned from twitter by summarisation," Association for Computational Linguistics (ACL), 2014.

[19] W. Kou, F. Li, and T. Baldwin, "Automatic labelling of topic models using word vectors and letter trigram vectors," in *Information Retrieval Technology: 11th Asia Information Retrieval Societies Conference, AIRS 2015, Brisbane, QLD, Australia, December 2-4, 2015. Proceedings 11*, Springer, 2015, pp. 253–264.

[20] S. Bhatia, J. H. Lau, and T. Baldwin, "Automatic labelling of topics with neural embeddings," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 953–963. [Online]. Available: https://aclanthology.org/C16-1091.

[21] D. Magatti, S. Calegari, D. Ciucci, and F. Stella, "Automatic labeling of topics," in *2009 Ninth International Conference on Intelligent Systems Design and Applications*, 2009, pp. 1227–1232. DOI: 10.1109/ISDA.2009.165.

[22] X.-L. Mao, Z.-Y. Ming, Z.-J. Zha, T.-S. Chua, H. Yan, and X. Li, "Automatic labeling hierarchical topics," ser. CIKM '12, Maui, Hawaii, USA: Association for Computing Machinery, 2012, pp. 2383–2386, ISBN: 9781450311564. DOI: 10.1145/2396761.2398646. [Online]. Available: https://doi.org/10.1145/2396761.2398646.

[23] J. H. Lau, K. Grieser, D. Newman, and T. Baldwin, "Automatic labelling of topic models," in *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, 2011, pp. 1536–1545.

[24] M. Allahyari and K. Kochut, "Automatic topic labeling using ontology-based topic models," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 2015, pp. 259–264. DOI: 10.1109/ICMLA.2015.88.

[25] I. Hulpus, C. Hayes, M. Karnstedt, and D. Greene, "Unsupervised graph-based topic labelling using dbpedia," in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, ser. WSDM '13, Rome, Italy: Association for Computing Machinery, 2013, pp. 465–474, ISBN: 9781450318693. DOI: 10.1145/2433396.2433454. [Online]. Available: https://doi.org/10.1145/2433396.2433454.

[26] E. Zosa, L. Pivovarova, M. Boggia, and S. Ivanova, "Multilingual topic labelling of news topics using ontological mapping," in *Advances in Information Retrieval*, M. Hagen, S. Verberne, C. Macdonald, *et al.*, Eds., Cham: Springer International Publishing, 2022, pp. 248–256, ISBN: 978-3-030-99739-7.

[27] A. Alokaili, N. Aletras, and M. Stevenson, "Automatic generation of topic labels," ser. SIGIR '20, Virtual Event, China: Association for Computing Machinery, 2020, pp. 1965–1968, ISBN: 9781450380164. DOI: 10.1145/3397271.3401185. [Online]. Available: https://doi.org/10.1145/3397271.3401185.

[28] C. Popa and T. Rebedea, "BART-TL: Weakly-supervised topic label generation," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online: Association for Computational Linguistics, Apr. 2021, pp. 1418–1425. DOI: 10.18653/v1/2021.eacl-main.121. [Online]. Available: https://aclanthology.org/2021.eacl-main.121.

[29] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. DOI: 10.18653/v1/D19-1410. [Online]. Available: https://aclanthology.org/D19-1410.

[30] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, *Bertscore: Evaluating text generation with bert*, 2020. arXiv: 1904.09675 [cs.CL].

[31] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, no. 1, pp. 36–45, Jan. 1966, ISSN: 0001-0782. DOI: 10.1145/365153.365168. [Online]. Available: https://doi.org/10.1145/365153.365168.

[32] T. Winograd, "Procedures as a representation for data in a computer program for understanding natural language," MASSACHUSETTS INST OF TECH CAMBRIDGE PROJECT MAC, Tech. Rep., 1971.

[33] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, R. L. Mercer, *et al.*, "The mathematics of statistical machine translation: Parameter estimation," 1993.

[34] D. Ferrucci, A. Levas, S. Bagchi, D. Gondek, and E. T. Mueller, "Watson: Beyond jeopardy!" *Artificial Intelligence*, vol. 199-200, pp. 93–105, 2013, ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2012.06.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370212000872.

[35] M. E. Peters, M. Neumann, M. Iyyer, *et al.*, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 2227–

2237. DOI: 10.18653/v1/N18-1202. [Online]. Available: https://aclanthology.org/N18-1202.

[36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[37] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

[38] Y. Liu, M. Ott, N. Goyal, *et al.*, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. arXiv: 1907.11692. [Online]. Available: http://arxiv.org/abs/1907.11692.

[39] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[40] *Introducing chatgpt*. [Online]. Available: https://openai.com/blog/chatgpt (visited on 03/25/2023).

[41] E. Collins and Z. Ghahramani, *Lamda: Our breakthrough conversation technology*, May 2021. [Online]. Available: https://blog.google/technology/ai/lamda/.

[42] S. Pichai, *An important next step on our ai journey*, Feb. 2023. [Online]. Available: https://blog.google/technology/ai/bard-google-ai-search-updates/.

[43] C. Badenes-Olmedo, J. L. Redondo-Garcia, and O. Corcho, "Distributing text mining tasks with librairy," in *Proceedings of the 2017 ACM Symposium on Document Engineering*, ser. DocEng '17, Valletta, Malta: Association for Computing Machinery, 2017, pp. 63–66, ISBN: 9781450346894. DOI: 10.1145/3103010.3121040. [Online]. Available: https://doi.org/10.1145/3103010.3121040.

[44] *Hugging face – the ai community building the future.* [Online]. Available: https://huggingface.co/ (visited on 03/25/2023).

[45] *Sentencetransformers documentation*. [Online]. Available: https://www.sbert.net/index.html (visited on 03/10/2023).

[46] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Nov. 2019. [Online]. Available: https://arxiv.org/abs/1908.10084.

[47] *Pretrained models - sentence-transformers documentation*. [Online]. Available: https://www.sbert.net/docs/pretrained_models.html#sentence-embedding-models/ (visited on 03/10/2023).

[48] *Sentence-transformers/all-mpnet-base-v2 · hugging face*, 2022. [Online]. Available: https://huggingface.co/sentence-transformers/all-mpnet-base-v2 (visited on 03/10/2023).

[49] *Sentence-transformers/all-minilm-l6-v2 · hugging face*. [Online]. Available: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2 (visited on 03/10/2023).

[50] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," *CoRR*, vol. abs/1806.03822, 2018. arXiv: 1806.03822. [Online]. Available: http://arxiv.org/abs/1806.03822.

[51] *Papers with code - squadv2 benchmark (question answering)*. [Online]. Available: https://paperswithcode.com/sota/question-answering-on-squad-v2 (visited on 03/17/2023).

[52] *Deepset/deberta-v3-large-squad2 · hugging face*. [Online]. Available: https://huggingface.co/deepset/deberta-v3-large-squad2 (visited on 03/17/2023).

[53] *Deepset/deberta-v3-base-squad2 · hugging face*. [Online]. Available: https://huggingface.co/deepset/deberta-v3-base-squad2 (visited on 03/17/2023).

[54] *Deepset/xlm-roberta-large-squad2 · hugging face*. [Online]. Available: https://huggingface.co/deepset/xlm-roberta-large-squad2 (visited on 03/17/2023).

[55] *Deepset/bert-large-uncased-whole-word-masking-squad2 · hugging face*. [Online]. Available: https://huggingface.co/deepset/bert-large-uncased-whole-word-masking-squad2 (visited on 03/17/2023).

[56] *Deepset/roberta-base-squad2-distilled · hugging face*. [Online]. Available: https://huggingface.co/deepset/roberta-base-squad2-distilled (visited on 03/17/2023).

[57] *Pygmalionai/pygmalion-6b · hugging face*. [Online]. Available: https://huggingface.co/PygmalionAI/pygmalion-6b (visited on 03/25/2023).

[58] *Facebook/blenderbot-400m-distill · hugging face*. [Online]. Available: https://huggingface.co/facebook/blenderbot-400M-distill (visited on 03/25/2023).

[59] *Microsoft/dialogpt-medium · hugging face*. [Online]. Available: https://huggingface.co/microsoft/DialoGPT-medium (visited on 03/25/2023).

[60] *Models - hugging face*. [Online]. Available: https://huggingface.co/models (visited on 03/25/2023).

Acknowledgements

Quiero expresar mi más profundo agradecimiento a mis tutores, Carlos y Óscar, por su orientación y apoyo a lo largo de mi tesis. Su experiencia, comentarios y compromiso han sido fundamentales para dar forma a este trabajo y elevar su calidad. Me siento realmente afortunada de haber tenido la oportunidad de aprender de ellos. Además, quiero agradecer a mi familia por su apoyo incondicional, aliento y confianza en mí en todo lo que hago. Su amor, paciencia, comprensión y ejemplo han sido una fuente constante de apoyo y ánimos para seguir trabajando para ser la mejor versión de mí cada día.

Appendix A

Results tables

		Q1: Which is the topic that contains the words … ?									Q2: What is the topic related to the words … ?									Q3: The words … are related to which common topic?								
		1 word			5 words			10 words			1 word			5 words			10 words			1 word			5 words			10 words		
	Conv. models	blenderbot-400M-distil	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT
Topics	sport hockey	0,17245698	0,085800119	0,361228764	0,678949177	0,20043838	0,712077796	0,667791069	0,118299924	0,822804451	0,263615519	0,085800126	0,294315249	0,678949177	0,245022565	0,724330723	0,667791069	0,245022565	0,82280457	0,022114797	0,206974104	0,373817146	0,667791069	0,235132843	0,669171214	0,667791128	0,351939678	0,822804511
	religion atheism	0,49848029	0,318726122	0,622302175	0,268774778	0,268382192	0,402470648	0,237399355	0,28774336	0,503110945	0,302259743	0,097775131	0,478190064	0,180191606	0,144547313	0,554925501	0,237399355	0,144547313	0,59698379	0,296922177	0,388167292	0,569720149	0,307034701	0,530157149	0,568023562	0,340552688	0,391716152	0,372681201
	science space	0,392073721	0,502478838	0,545285225	0,392073721	0,233378083	0,581333756	0,2989932	0,333704323	0,682387054	0,444410563	0,502478719	0,590685904	0,392073721	0,373763055	0,616905808	0,397340417	0,225318253	0,682387114	0,457178921	0,112608358	0,701213717	0,392073721	0,223453075	0,595674992	0,443360925	0,225318164	0,682387233
	science medicine	0,213946685	0,293799013	0,570342004	0,414733768	0,035892993	0,613653839	0,101762079	0,104785167	0,731180787	0,330108464	0,251149714	0,467416763	0,414733768	-0,010164938	0,629934609	0,154717654	-0,010164938	0,67343545	0,102340356	0,266858101	0,486507356	0,324584454	-0,010164935	0,604890108	0,419126809	-0,010164932	0,69918289
	politics misc	0,195432395	0,270650566	0,526503086	-0,000970341	0,288078964	0,566280007	-0,036618955	0,163024396	0,600974858	0,195432395	0,190067679	0,436191618	-0,01940966	0,190067679	0,519822001	-0,036618955	0,190067679	0,519810438	-0,04625839	0,406030416	0,482934088	-0,014507763	0,214465901	0,458757401	-0,018152099	0,244898856	0,696976244
	computer mac hardware	0,330494583	0,293022871	0,463650048	0,188131511	0,29858467	0,484241664	-0,01790693	0,63967061	0,594033599	0,330494553	0,293022871	0,512844324	0,188131511	-0,052205443	0,578874588	-0,075140305	-0,052205443	0,654822588	0,360641599	0,293022811	0,678993702	0,339022964	0,555669427	0,639931738	0,339022994	0,642169774	0,750000715
	politics mideast	0,176571727	0,09407717	0,274707735	0,191067159	0,11467094	0,536305547	0,18017149	0,121043049	0,560694993	0,176571742	0,09407714	0,260530442	0,308806449	0,11467088	0,598001182	0,18017149	0,11467088	0,689873695	0,192945346	0,110027015	0,291417599	0,200759217	0,262622148	0,609008193	0,445147336	0,114280216	0,872331142
	computer ibm hardware	0,041777141	0,146203637	0,231412798	0,120998971	0,320700943	0,423096001	-0,059531681	0,295660317	0,495372862	0,051721968	0,146203622	0,10900747	0,253789693	0,003184831	0,473391026	0,148587137	0,115117535	0,549951911	0,125904053	0,146203592	0,203755572	0,058344807	0,19149107	0,478031933	0,140041083	0,352395266	0,495372802
	for sale	0,170681715	0,155119985	0,112167671	-0,041566409	0,241281599	0,189814433	0,090107366	0,19016844	0,187696666	0,108716719	0,197042212	0,25194636	-0,024151515	0,026592733	0,308535814	-0,020213399	0,223840922	0,23955071	0,147700012	0,196251184	0,322404027	-0,065779112	0,104629599	0,333043635	-0,059302293	0,163481742	0,288431585
	science electronics	0,050484382	-0,014091416	0,058079284	0,254096955	-0,030010957	0,552863657	0,086560309	0,217775881	0,790198088	0,076422192	-0,03824193	0,099258639	0,141136885	-0,03001089	0,537722468	0,056047559	-0,03001089	0,769300342	0,005884372	0,060250659	0,040058851	0,254096955	0,27718371	0,441956103	0,013320578	0,303878695	0,790197968
	computer windows misc	0,025842385	0,28868252	0,4982889	0,12472184	0,042588398	0,49372074	0,022691274	0,377550066	0,557667553	0,036112934	0,199065536	0,309096694	0,031464841	0,199065611	0,575948	0,20015344	-0,032611303	0,504279554	0,312496662	0,199065492	0,451176703	0,143714949	0,461758435	0,455995172	0,143714964	0,397323251	0,465785325
	motor motorcycle	0,304482788	0,336321771	0,431906223	0,301933885	0,016140021	0,320436418	0,001543007	0,016140057	0,55488193	0,535691321	0,329999983	0,366602331	-0,050301414	0,016140014	0,505816221	0,001543007	0,016140014	0,638510942	0,324788809	0,395962596	0,458796948	-0,031872682	0,016139977	0,371625245	0,670521498	0,224839076	0,66650784
	sport baseball	0,161658436	0,073340267	0,364671648	0,379656106	0,276418149	0,645073593	0,345497459	0,353214651	0,847991228	0,24137798	0,073340297	0,292123675	0,289812237	0,163075298	0,603144765	0,364194274	0,163075298	0,635453105	0,012119067	0,198236987	0,350157231	0,161658436	0,270022571	0,576705217	0,349394023	0,355805665	0,872383177
	religion christian	0,42370975	0,320748031	0,569514453	0,512571394	0,156869113	0,578466892	0,257555544	0,305597752	0,801881552	0,308159143	0,088649511	0,482132792	0,434937954	0,12215516	0,627174675	0,394068539	0,318851084	0,53027159	0,266128629	0,339308977	0,607395709	0,512571394	0,228451282	0,801881611	0,523530364	0,216869831	0,830949068
	politics guns	0,321564287	0,282257646	0,612956643	0,329971015	0,19679898	0,664190829	-0,003640767	0,158294931	0,706830263	0,612178504	0,284804791	0,534287512	-0,00840255	0,196798921	0,647349656	0,347807467	0,189203382	0,573405802	0,415679157	0,492646158	0,864738703	-0,00840255	0,280908287	0,917803049	0,346214801	0,189203426	0,698238313
	computer graphics	0,14736557	0,218678176	0,387340933	0,042030804	0,020478372	0,580949843	0,331439018	0,119503327	0,453237712	0,229187459	0,039627228	0,320134878	0,106684856	0,020478351	0,450405478	0,046456635	0,020478351	0,57693404	0,194999218	0,205066651	0,36371544	0,042030804	0,113985352	0,525060177	0,422228992	0,07869029	0,872970939
	motor autos	0,209309518	0,255402535	0,451042295	0,214226395	0,209855452	0,468370467	0,091886058	0,350901097	0,538020134	0,514179111	0,255402505	0,421530426	0,305929869	0,380782187	0,489159107	0,091886058	0,380782187	0,480545908	0,258245379	0,299046636	0,531019926	0,092310205	0,218954355	0,48413533	0,092310198	0,314151347	0,568026125
	religion misc	0,360829711	0,357301891	0,56375432	0,175573856	0,335864127	0,573547065	0,318217099	0,361007005	0,612212062	0,232523382	0,161154628	0,470437199	0,206974387	0,185483158	0,577567995	0,318217099	0,23054561	0,458695114	0,202267691	0,319806784	0,510488808	0,34887135	0,223342195	0,57714802	0,318217099	0,184710413	0,753349543
	computer windows x	0,020045783	0,238506198	0,4296408	0,120956019	0,34331429	0,345453465	0,086787529	0,206449643	0,424537897	0,02368124	0,14985691	0,180101588	0,194686785	0,153060079	0,370075226	-0,067412682	-0,026326194	0,396125913	0,233775809	0,149856895	0,365944982	0,120956019	0,547588885	0,275777727	0,158811927	0,149856851	0,555624247
	science crypt	0,162083477	0,138862655	0,284100652	0,094704352	0,189268142	0,379152536	0,151000857	0,059734464	0,312836826	0,207069561	0,138862669	0,290412992	0,199771881	0,093757398	0,406672597	0,223705977	0,099377587	0,334300429	0,167933911	0,197996467	0,238126338	0,094704352	0,084000751	0,368635118	0,094704352	0,084000714	0,383553326
AVERAGE SIMILARITY		**0,218964566**	**0,23279443**	**0,417944783**	**0,238131748**	**0,187949643**	**0,505575005**	**0,157585219**	**0,239013423**	**0,588927573**	**0,260995725**	**0,177006967**	**0,358362346**	**0,211290524**	**0,126813198**	**0,539787872**	**0,181535092**	**0,126285995**	**0,566372115**	**0,202690379**	**0,249169359**	**0,44461915**	**0,196998165**	**0,251489607**	**0,537662777**	**0,292527868**	**0,248768224**	**0,655387771**
SCORE		**18**	**17**	**8**	**16**	**22**	**6**	**25**	**15**	**2**	**11**	**24**	**9**	**19**	**26**	**4**	**23**	**27**	**3**	**20**	**13**	**7**	**21**	**12**	**5**	**10**	**14**	**1**
AGGREGATED SCORE		129									146									103								
RANKING BY AGGREGATED SCORE		2									3									1								

Table A.1: Question structure selection evaluation

		1 word			2 words			3 words			4 words			5 words			6 words			7 words			8 words			9 words			10 words		
	Conv. models	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT
Topics	sport hockey	0,022115	0,206974	0,373817	0,240472	0,35194	0,683555	0,162884	0,21692	0,683555	0,167091	0,57072	0,683555	0,667791	0,235133	0,669171	0,667791	0,35194	0,822805	0,667791	0,35194	0,822805	0,667791	0,35194	0,822805	0,667791	0,35194	0,683555	0,667791	0,35194	0,822805
	religion atheism	0,296922	0,388167	0,56972	0,29212	0,459821	0,710449	0,29212	0,234119	0,710449	0,307035	0,530157	0,568024	0,231467	0,530157	0,710449	0,231467	0,388167	0,66307	0,307035	0,40219	0,503111	0,241285	0,530157	0,710449	0,340553	0,391716	0,372681			
	science space	0,457179	0,112608	0,701214	0,454496	0,481966	0,682387	0,392074	0,373763	0,682387	0,392074	0,020608	0,682387	0,392074	0,223453	0,595675	0,392074	0,263015	0,682387	0,392074	0,325149	0,682387	0,392074	0,325149	0,682387	0,392074	0,302213	0,682387	0,443361	0,225318	0,682387
	science medicine	0,10234	0,266858	0,486507	0,179477	0,240049	0,669183	0,177037	0,202003	0,669183	0,324584	0,202003	0,669183	0,324584	-0,01016	0,60489	0,324584	-0,01016	0,669183	0,308426	-0,01016	0,669183	0,210292	-0,01016	0,669183	0,142841	0,302848	0,669183	0,419127	-0,01016	0,669183
	politics misc	-0,04626	0,40603	0,482934	-0,00091	0,235198	0,68987	0,05262	0,257036	0,696976	-0,01451	0,214466	0,696976	-0,01451	0,214466	0,458757	-0,01451	0,217398	0,661725	-0,01451	0,030832	0,696976	-0,01451	0,213912	0,696976	0,036937	0,244899	0,661725	-0,01815	0,244899	0,696976
	computer mac hardware	0,360642	0,293023	0,678994	0,189043	0,293023	0,375127	0,199351	0,524383	0,34712	0,330495	0,392794	0,375127	0,339023	0,555669	0,639932	0,339023	0,555669	0,750001	0,339023	0,555669	0,576266	0,339023	0,64217	0,617081	0,339023	0,646929	0,750001	0,339023	0,64217	0,750001
	politics misc	0,192945	0,110027	0,291418	0,198413	0,347769	0,737193	0,180856	0,240676	0,675998	0,14476	0,181686	0,68376	0,200759	0,263622	0,669908	0,191067	0,263622	0,868063	0,180171	0,121043	0,816431	0,180171	0,11428	0,776419	0,180856	0,11428	0,794242	0,445147	0,11428	0,872331
	computer ibm hardware	0,125904	0,146204	0,203756	0,112149	0,146204	0,203045	0,052511	0,146204	0,429792	0,058345	0,066404	0,685402	0,058345	0,191491	0,478032	0,058345	0,346166	0,495373	0,06835	0,346166	0,74048	0,105234	0,352395	0,685402	0,050571	0,352395	0,74048	0,140041	0,352395	0,495373
	for sale	0,1477	0,196251	0,322404	0,205518	0,369535	0,161264	0,217266	0,369535	0,161264	0,132628	0,334268	0,312039	-0,06578	0,10463	0,333044	0,109066	0,369535	0,288432	0,204157	0,026593	0,285552	0,204356	0,026593	0,190212	0,248695	0,163482	0,288552	-0,0593	0,163482	0,288432
	science electronics	0,005884	0,060251	0,040059	0,254097	0,276064	0,812445	0,254097	0,261983	0,756294	0,254097	0,277184	0,690404	0,254097	0,277184	0,441956	0,020791	0,277184	0,690404	-0,00319	0,277184	0,799603	0,254097	0,258817	0,799603	0,013321	0,292277	0,790198	0,013321	0,303879	0,790198
	computer windows misc	0,312497	0,199065	0,451177	0,173574	0,199065	0,465785	0,199065	0,199065	0,386647	0,007838	0,461758	0,593516	0,143715	0,461758	0,455995	0,153385	0,461758	0,486801	0,143715	0,383542	0,593516	0,020129	0,38816	0,593516	0,143715	0,387937	0,479202	0,143715	0,397323	0,465785
	motor motorcycle sport	0,324789	0,395963	0,458797	0,324789	0,277695	0,458797	0,324789	0,277695	0,348403	0,631454	0,277695	0,711446	-0,03187	0,01614	0,371625	-0,03187	0,01614	0,384717	0,301934	0,01614	0,63974	0,301934	0,01614	0,63974	0,324789	0,01614	0,548275	0,670521	0,224839	0,666508
	baseball	0,012119	0,198237	0,350157	0,245207	0,345664	0,673092	0,153581	0,163075	0,673092	0,153581	0,163075	0,626936	0,161658	0,270023	0,576705	0,13636	0,292913	0,590647	0,639771	0,345664	0,847991	0,639771	0,345664	0,847991	0,364194	0,355806	0,847991	0,349394	0,355806	0,872383
	religion christian	0,266129	0,339309	0,607396	0,26588	0,095182	0,813691	0,375227	0,228451	0,785618	0,512571	0,228451	0,813691	0,512571	0,228451	0,801882	0,512571	0,214066	0,830949	0,394069	0,218851	0,786605	0,394069	0,318851	0,830949	0,434938	0,337923	0,755603	0,52353	0,21687	0,830949
	politics guns	0,415679	0,492646	0,864739	0,321564	0,584732	0,600266	0,408117	0,280908	0,833261	0,329971	0,584732	0,748153	-0,0084	0,280908	0,917803	0,027365	0,280908	0,616599	0,349249	0,189203	0,700616	-0,00364	0,189203	0,691809	0,035742	0,189203	0,691809	0,346215	0,189203	0,698238
	computer graphics	0,194999	0,205067	0,363715	0,083944	0,154878	0,533035	0,138629	0,049772	0,700328	0,116422	0,113985	0,862105	0,042031	0,113985	0,525106	0,116422	0,113985	0,617627	0,042031	0,298833	0,563121	0,097845	0,113985	0,857774	0,116422	0,298833	0,698066	0,422229	0,07869	0,872971
	motor autos	0,258245	0,295047	0,53102	0,09231	0,431007	0,644577	0,09231	0,361143	0,550754	0,09231	0,420252	0,564576	0,09231	0,218954	0,484135	0,09231	0,218705	0,659539	0,235241	0,309517	0,545735	0,09231	0,415482	0,545735	0,09231	-0,0219	0,545735	0,09231	0,314151	0,568025
	religion misc	0,202268	0,319807	0,510489	0,206974	0,340003	0,751578	0,206974	0,340003	0,737178	0,206974	0,223342	0,656573	0,348871	0,223342	0,577148	0,169204	0,13471	0,525341	0,348871	0,13471	0,661616	0,348871	0,13471	0,684105	0,348871	0,13471	0,590501	0,318217	0,13471	0,75335
	computer windows x	0,233776	0,149857	0,365945	0,127185	0,149857	0,438303	0,120956	0,149857	0,347129	0,120956	0,412955	0,502864	0,547589	0,275778	0,223618	0,15306	0,320285	-0,0181	0,149857	0,264522	0,006562	0,149857	0,264522	-0,01068	0,440164	0,272668	0,158812	0,149857	0,555624	
	windows crypt	0,165934	0,197995	0,258126	0,085128	0,145295	0,260498	0,094704	0,00807	0,407708	0,084001	0,413654	0,084001	0,358635	0,094704	0,358635	0,094704	0,075939	0,310059	0,094704	0,083881	0,407708	0,094704	0,084001	0,383553	0,094704	0,083881	0,234415	0,094704	0,084001	0,383553
	AVERAGE SIMILARITY	0,202269	0,249169	0,444619	0,202532	0,296247	0,583162	0,203692	0,243426	0,579157	0,239985	0,288027	0,63414	0,196998	0,25149	0,587663	0,182039	0,262103	0,599069	0,245272	0,234639	0,638346	0,231906	0,244167	0,639141	0,21292	0,278706	0,621752	0,292528	0,248768	0,655388
	SCORE	27	17	10	28	11	8	26	21	7	22	13	4	29	16	9	30	15	6	19	23	3	24	20	2	25	14	5	12	18	1
	AGGREGATED SCORE	54			47			54			39			54			30			45			46			44			31		
	RANKING BY AGGREGATED SCORE	8			6			8			2			8			7			4			5			3			1		

Table A.2: Number of words selection evaluation

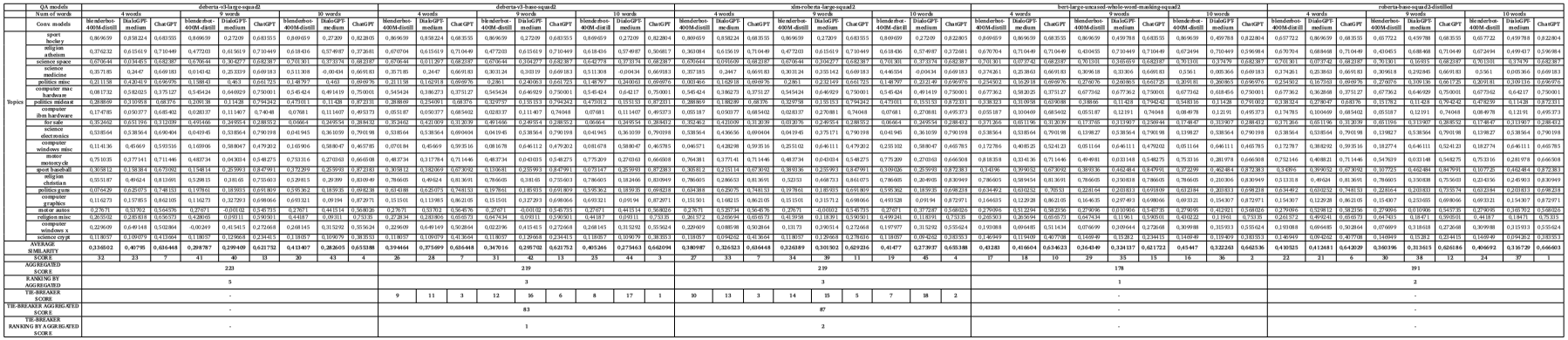


Table A.3: Question-Answering models selection evaluation

		Words	4			9			10			AVERAGE TOPIC SIMILARITY	TOPIC RANKING
		Conv. models	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT		
Topics	0	sport hockey	0,667791	0,57072	0,683555	0,667791	0,35194	0,683555	0,667791	0,35194	0,822805	**0,607543**	**1**
	1	religion atheism	0,307035	0,530157	0,710449	0,241285	0,530157	0,710449	0,340553	0,391716	0,372681	**0,459387**	**4**
	2	science space	0,392074	0,020608	0,682387	0,392074	0,302213	0,682387	0,443361	0,225318	0,682387	**0,424757**	**7**
	3	science medicine	0,260202	0,202003	0,669183	0,142841	0,302848	0,669183	0,419127	-0,01016	0,669183	**0,369378**	**13**
	4	politics_misc	-0,01451	0,214466	0,696976	0,036937	0,244899	0,661725	-0,01815	0,244899	0,696976	**0,307135**	**16**
	5	computer mac hardware	0,330495	0,392794	0,375127	0,339023	0,646929	0,750001	0,339023	0,64217	0,750001	**0,507285**	**3**
	6	politics mideast	0,14476	0,181686	0,68376	0,180856	0,11428	0,794242	0,445147	0,11428	0,872331	**0,392371**	**10**
	7	computer ibm hardware	0,058345	0,066404	0,685402	0,050571	0,352395	0,74048	0,140041	0,352395	0,495373	**0,326823**	**15**
	8	for sale	0,132628	0,334268	0,312039	0,248695	0,163482	0,288552	-0,0593	0,163482	0,288432	**0,208031**	**19**
	9	science electronics	0,254097	0,277184	0,690404	0,013321	0,292277	0,790198	0,013321	0,303879	0,790198	**0,380542**	**12**
	10	computer windows misc	0,007838	0,461758	0,593516	0,143715	0,387937	0,479202	0,143715	0,397323	0,465785	**0,34231**	**14**
	11	motor motorcycle	0,631454	0,277695	0,711446	0,324789	0,01614	0,548275	0,670521	0,224839	0,666508	**0,452407**	**6**
	12	sport baseball	0,153581	0,163075	0,626936	0,364194	0,355806	0,847991	0,349394	0,355806	0,872383	**0,454352**	**5**
	13	religion christian	0,512571	0,228451	0,813691	0,434938	0,337923	0,755603	0,52353	0,21687	0,830949	**0,51717**	**2**
	14	politics guns	0,329971	0,584732	0,748153	0,035742	0,189203	0,691809	0,346215	0,189203	0,698238	**0,423696**	**8**
	15	computer graphics	0,116422	0,113985	0,862105	0,116422	0,298833	0,698066	0,422229	0,07869	0,872971	**0,397747**	**9**
	16	motor autos	0,09231	0,420252	0,564576	0,09231	-0,0219	0,545735	0,09231	0,314151	0,568026	**0,296419**	**17**
	17	religion misc	0,206974	0,223342	0,656573	0,348871	0,18471	0,590501	0,318217	0,18471	0,75335	**0,38525**	**11**
	18	computer windows x	0,120956	0,412955	0,502864	-0,01068	0,440164	0,272668	0,158812	0,149857	0,555624	**0,289246**	**18**
	19	science crypt	0,094704	0,084001	0,413664	0,094704	0,083881	0,234415	0,094704	0,084001	0,383553	**0,174181**	**20**
AVERAGE SIMILARITY			**0,239985**	**0,288027**	**0,63414**	**0,21292**	**0,278706**	**0,621752**	**0,292528**	**0,248768**	**0,655388**		
SCORE			**8**	**5**	**2**	**9**	**6**	**3**	**4**	**7**	**1**		
AGGREGATED SCORE			**15**			**18**			**12**				
NUM. RANKING BY AGGREGATED SCORE			**2**			**3**			**1**				
CONV. MODELS RANKING BY SCORE			**1**	**2**	**3**	**1**	**2**	**3**	**2**	**1**	**3**		

Table A.4: Conversational models evaluation using "all-mpnet-base-v2" embeddings

		Words	4			9			10			AVERAGE TOPIC SIMILARITY	TOPIC RANKING
		Conv. models	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT		
Topics	0	sport hockey	0,765336	0,638691	0,744449	0,765336	0,345455	0,744449	0,765336	0,345455	0,898747	**0,668139**	1
	1	religion atheism	0,40964	0,463976	0,648731	0,334879	0,463976	0,648731	0,452754	0,482089	0,575918	**0,497855**	5
	2	science space	0,542915	0,048859	0,642193	0,542915	0,293662	0,642193	0,383065	0,321972	0,642193	**0,451108**	7
	3	science medicine	0,314172	0,206525	0,644639	0,270071	0,46696	0,644639	0,480386	0,120559	0,644639	**0,421399**	8
	4	politics_misc	0,136562	0,209315	0,572057	0,040087	0,264367	0,53184	0,038668	0,264367	0,572057	**0,292147**	15
	5	computer mac hardware	0,387205	0,367129	0,368116	0,414116	0,681641	0,802606	0,414116	0,70257	0,802607	**0,548901**	2
	6	politics mideast	0,050109	0,034946	0,314869	0,07684	0,215507	0,421431	0,211653	0,215507	0,47598	**0,224094**	20
	7	computer ibm hardware	0,03769	-0,05368	0,609845	-0,00312	0,158852	0,726169	0,100969	0,158852	0,453806	**0,243264**	18
	8	for sale	0,232234	0,398449	0,450034	0,366739	0,216483	0,379015	0,013035	0,216483	0,38691	**0,295487**	14
	9	science electronics	0,382243	0,323593	0,670174	0,036364	0,31302	0,738929	0,036364	0,460402	0,738929	**0,411113**	9
	10	computer windows misc	0,007184	0,361778	0,443303	0,064919	0,238528	0,441178	0,064919	0,238689	0,273379	**0,237097**	19
	11	motor motorcycle	0,721538	0,335078	0,7445	0,332579	0,136295	0,71691	0,718464	0,259099	0,726304	**0,521196**	3
	12	sport baseball	0,158571	0,298066	0,751949	0,370579	0,341993	0,939255	0,414337	0,341993	0,844743	**0,49572**	6
	13	religion christian	0,53444	0,19028	0,763165	0,464925	0,357027	0,795788	0,554087	0,109725	0,800431	**0,507763**	4
	14	politics guns	0,279111	0,473431	0,705349	-0,02365	0,094899	0,557252	0,291958	0,094899	0,656528	**0,347752**	11
	15	computer graphics	0,036431	0,137318	0,883062	0,036431	0,252962	0,635572	0,54562	0,180928	0,82836	**0,392965**	10
	16	motor autos	0,120362	0,505477	0,561865	0,120362	0,085158	0,59044	0,120362	0,362795	0,543897	**0,334524**	12
	17	religion misc	0,206035	0,251791	0,52233	0,396553	0,13103	0,461674	0,247797	0,13103	0,605732	**0,328219**	13
	18	computer windows x	0,072568	0,453458	0,460816	0,017245	0,314907	0,392636	0,133869	0,119623	0,430332	**0,266161**	16
	19	science crypt	0,217361	0,096908	0,511226	0,217361	0,061867	0,383692	0,217361	0,096908	0,526853	**0,258838**	17
AVERAGE SIMILARITY			**0,280585**	**0,287069**	**0,600634**	**0,242076**	**0,271729**	**0,60972**	**0,310256**	**0,261197**	**0,621417**		
SCORE			**6**	**5**	**3**	**9**	**7**	**2**	**4**	**8**	**1**		
AGGREGATED SCORE			**14**			**18**			**13**				
NUM. RANKING BY AGGREGATED SCORE			**2**			**3**			**1**				
CONV. RANKING BY SCORE			**3**	**2**	**1**	**3**	**2**	**1**	**2**	**3**	**1**		

Table A.5: Conversational models evaluation using "MiniLM-L6-v2" embeddings

		QA Model	deberta-v3-base-squad2								
		Words	4			9			10		
		Conv. Model	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT
Topics	0	sport hockey	0,869659	0,858224	0,683555	0,869659	0,27209	0,683555	0,869659	0,27209	0,822804
	1	religion atheism	0,670704	0,615619	0,710449	0,477203	0,615619	0,710449	0,618436	0,574987	0,506817
	2	science space	0,670644	0,011297	0,682387	0,670644	0,304277	0,682387	0,642778	0,373374	0,682387
	3	science medicine	0,357185	0,2447	0,669183	0,303124	0,30319	0,669183	0,511308	-0,00434	0,669183
	4	politics misc	0,211158	0,162918	0,696976	0,2861	0,240063	0,661725	0,148797	0,240063	0,696976
	5	computer mac hardware	0,545424	0,386273	0,375127	0,545424	0,646929	0,750001	0,545424	0,64217	0,750001
	6	politics mideast	0,288869	0,254091	0,68376	0,329757	0,155153	0,794242	0,473012	0,155153	0,872331
	7	computer ibm hardware	0,055187	0,050377	0,685402	0,028337	0,111407	0,74048	0,07681	0,111407	0,495373
	8	for sale	0,352462	0,421009	0,312039	0,491466	0,249554	0,288552	0,06664	0,249554	0,288432
	9	science electronics	0,538564	0,538564	0,690404	0,041945	0,538564	0,790198	0,041945	0,361059	0,790198
	10	computer windows misc	0,070184	0,45669	0,593516	0,081678	0,646112	0,479202	0,081678	0,588047	0,465785
	11	motor motorcycle	0,483734	0,317784	0,711446	0,483734	0,043035	0,548275	0,775209	0,270363	0,666508
	12	sport baseball	0,305812	0,382069	0,673092	0,130681	0,255993	0,847991	0,073147	0,255993	0,872383
	13	religion christian	0,786605	0,49624	0,813691	0,786605	0,38165	0,755603	0,786605	0,182466	0,830949
	14	politics guns	0,634388	0,625075	0,748153	0,197861	0,185935	0,691809	0,595362	0,185935	0,698238
	15	computer graphics	0,151501	0,113985	0,862105	0,151501	0,327293	0,698066	0,693321	0,09194	0,872971
	16	motor autos	0,27671	0,53702	0,564576	0,27671	-0,00102	0,545735	0,27671	0,441514	0,568026
	17	religion misc	0,272834	0,283806	0,656573	0,647434	0,09311	0,590501	0,44187	0,09311	0,75335
	18	computer windows x	0,229609	0,649149	0,502864	0,022396	0,415415	0,272668	0,268145	0,315292	0,555624
	19	science crypt	0,118057	0,109079	0,413664	0,118057	0,129668	0,234415	0,118057	0,109079	0,383553
AVERAGE SIMILARITY			**0,394464**	**0,375699**	**0,636448**	**0,347016**	**0,295702**	**0,621752**	**0,405246**	**0,275463**	**0,662094**
SCORE			17	18	5	21	26	8	16	27	3
NUM. SCORE			5	6	2	7	8	3	4	9	1
NUM. AGGREGATED SCORE			13			18			14		
NUM. RANKING BY NUM. AGGREGATED SCORE			1			3			2		
AGGREGATED SCORES			141								
QA RANKING BY AGGREGATED SCORES			3								

		QA Model	bert-large-uncased-whole-word-masking-squad2								
		Words	4			9			10		
		Conv. Model	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT
Topics	0	sport hockey	0,869659	0,869659	0,683555	0,869659	0,459788	0,683555	0,869659	0,459788	0,822804
	1	religion atheism	0,670704	0,710449	0,710449	0,430455	0,710449	0,710449	0,672494	0,710449	0,596984
	2	science space	0,701301	0,073742	0,682387	0,701301	0,365659	0,682387	0,701301	0,37479	0,682387
	3	science medicine	0,374261	0,253863	0,669183	0,309618	0,33306	0,669183	0,5561	0,005366	0,669183
	4	politics misc	0,254502	0,162918	0,696976	0,276076	0,260865	0,661725	0,209181	0,260865	0,696976
	5	computer mac hardware	0,677362	0,582025	0,375127	0,677362	0,677362	0,750001	0,677362	0,618456	0,750001
	6	politics mideast	0,338323	0,310958	0,639088	0,38866	0,11428	0,794242	0,548316	0,11428	0,791002
	7	computer ibm hardware	0,055187	0,100469	0,685402	0,055187	0,12191	0,74048	0,084978	0,12191	0,495373
	8	for sale	0,371266	0,651196	0,312039	0,173765	0,313907	0,256944	0,174847	0,313907	0,288432
	9	science electronics	0,538564	0,538564	0,790198	0,139827	0,538564	0,790198	0,139827	0,538564	0,790198
	10	computer windows misc	0,172786	0,408525	0,524123	0,051164	0,646111	0,479202	0,051164	0,646111	0,465785
	11	motor motorcycle	0,818358	0,334136	0,711446	0,494981	0,033148	0,548275	0,753316	0,281978	0,666508
	12	sport baseball	0,34396	0,399052	0,673092	0,389336	0,462484	0,847991	0,372299	0,462484	0,872383
	13	religion christian	0,786605	0,589454	0,813691	0,786605	0,350838	0,786605	0,786605	0,210306	0,830949
	14	politics guns	0,634492	0,630252	0,70553	0,228164	0,203833	0,691809	0,632384	0,203833	0,698238
	15	computer graphics	0,164635	0,122928	0,862105	0,164635	0,297493	0,698066	0,693321	0,154307	0,872971
	16	motor autos	0,279096	0,512294	0,582356	0,279096	0,010906	0,545735	0,279095	0,412921	0,568026
	17	religion misc	0,265503	0,265694	0,656573	0,647434	0,11961	0,590501	0,430222	0,11961	0,75335
	18	computer windows x	0,193088	0,696485	0,511434	0,076699	0,309644	0,272668	0,309988	0,315933	0,555624
	19	science crypt	0,146949	0,119409	0,407708	0,146949	0,15282	0,234415	0,146949	0,119409	0,383553
AVERAGE SIMILARITY			**0,43283**	**0,416604**	**0,634623**	**0,364349**	**0,324137**	**0,621722**	**0,45447**	**0,322263**	**0,662536**
SCORE			11	12	6	19	22	9	10	23	2
NUM. SCORE			5	6	2	7	8	3	4	9	1
NUM. AGGREGATED SCORE			13			18			14		
NUM. RANKING BY NUM. AGGREGATED SCORE			1			3			2		
AGGREGATED SCORES			114								
QA RANKING BY AGGREGATED SCORES			1								

		QA Model	roberta-base-squad2-distilled									AVERAGE TOPIC SIMILARITY	TOPIC RANKING
		Words	4			9			10				
		Conv. Model	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT		
Topics	0	sport hockey	0,657722	0,869659	0,683555	0,657722	0,459788	0,683555	0,657722	0,459788	0,822804	**0,694138**	**1**
	1	religion atheism	0,670704	0,688468	0,710449	0,430455	0,688468	0,710449	0,672494	0,499437	0,596984	**0,633003**	**2**
	2	science space	0,701301	0,073742	0,682387	0,701301	0,16935	0,682387	0,701301	0,37479	0,682387	**0,535347**	**5**
	3	science medicine	0,374261	0,253863	0,669183	0,309618	0,292845	0,669183	0,5561	0,005366	0,669183	**0,42082**	**13**
	4	politics misc	0,254502	0,167363	0,696976	0,276076	0,309136	0,661725	0,209181	0,309136	0,696976	**0,385405**	**15**
	5	computer mac hardware	0,677362	0,362868	0,375127	0,677362	0,646929	0,750001	0,677362	0,64217	0,750001	**0,61226**	**4**
	6	politics mideast	0,338324	0,278047	0,68376	0,151782	0,11428	0,794242	0,478259	0,11428	0,872331	**0,43966**	**12**
	7	computer ibm hardware	0,174785	0,100469	0,685402	0,055187	0,12191	0,74048	0,084978	0,12191	0,495373	**0,273932**	**19**
	8	for sale	0,371266	0,651196	0,312039	0,651196	0,313907	0,288552	0,174847	0,313907	0,288432	**0,331161**	**18**
	9	science electronics	0,538564	0,538564	0,790198	0,139827	0,538564	0,790198	0,139827	0,538564	0,790198	**0,516313**	**6**
	10	computer windows misc	0,172787	0,388292	0,593516	0,182774	0,646111	0,524123	0,182774	0,646111	0,465785	**0,396672**	**14**
	11	motor motorcycle	0,752146	0,408821	0,711446	0,547639	0,033148	0,548275	0,753316	0,281978	0,666508	**0,505389**	**8**
	12	sport baseball	0,34396	0,399052	0,673092	0,107725	0,462484	0,847991	0,107725	0,462484	0,872383	**0,477672**	**9**
	13	religion christian	0,513318	0,49624	0,813691	0,786605	0,350838	0,755603	0,234356	0,245903	0,830949	**0,621836**	**3**
	14	politics guns	0,634492	0,630252	0,748153	0,228164	0,203833	0,735574	0,632384	0,203833	0,698238	**0,515045**	**7**
	15	computer graphics	0,154307	0,122928	0,862105	0,154307	0,253655	0,698066	0,693321	0,154307	0,872971	**0,44293**	**10**
	16	motor autos	0,279096	0,529812	0,582356	0,279096	0,010906	0,545735	0,279095	0,365702	0,568026	**0,385012**	**16**
	17	religion misc	0,261572	0,499241	0,656573	0,647435	0,18471	0,590501	0,44187	0,18471	0,75335	**0,44078**	**11**
	18	computer windows x	0,193088	0,696485	0,502864	0,076699	0,318618	0,272668	0,309988	0,315933	0,555624	**0,359803**	**17**
	19	science crypt	0,146949	0,094262	0,407708	0,146949	0,15282	0,234415	0,146949	0,094262	0,383553	**0,199987**	**20**
AVERAGE SIMILARITY			**0,410525**	**0,412481**	**0,642029**	**0,360396**	**0,313615**	**0,626186**	**0,406692**	**0,316729**	**0,666603**		
SCORE			14	13	4	20	25	7	15	24	1		
NUM. SCORE			5	4	2	7	9	3	6	8	1		
NUM. AGGREGATED SCORE			11			19			15				
NUM. RANKING BY NUM. AGGREGATED SCORE			1			3			2				
AGGREGATED SCORES			123										
QA RANKING BY AGGREGATED SCORES			2										

Table A.6: Question-Answering models evaluation using "all-mpnet-base-v2" embeddings

		QA Model	deberta-v3-base-squad2									bert-large-uncased-whole-word-masking-squad2									roberta-base-squad2-distilled									AVERAGE TOPIC SIMILARITY	TOPIC RANKING
		Words	4			9			10			4			9			10			4			9			10				
		Conv. Model	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT	blenderbot-400M-distill	DialoGPT-medium	ChatGPT		
Topics	0	sport hockey	0,94868	0,856544	0,744449	0,94868	0,367009	0,744449	0,94868	0,367009	0,898747	0,94868	0,94868	0,744449	0,94868	0,452377	0,744449	0,94868	0,452377	0,898747	0,848595	0,94868	0,744449	0,848595	0,452377	0,744449	0,848595	0,452377	0,898747	**0,766601119**	1
	1	religion atheism	0,58622	0,629044	0,648731	0,390042	0,629044	0,648731	0,669699	0,592481	0,606489	0,58622	0,648731	0,648731	0,31792	0,648731	0,648731	0,632336	0,648731	0,647205	0,58622	0,607058	0,648731	0,31792	0,607058	0,648731	0,632336	0,503992	0,647205	**0,593594996**	4
	2	science space	0,756966	0,038573	0,642193	0,756966	0,323231	0,642193	0,696681	0,468508	0,642193	0,809825	0,160773	0,642193	0,809825	0,404021	0,642193	0,809825	0,475967	0,642193	0,809825	0,160773	0,642193	0,809825	0,100136	0,642193	0,809825	0,475967	0,642193	**0,572490846**	5
	3	science medicine	0,355034	0,218767	0,644639	0,385296	0,372664	0,644639	0,606173	0,15933	0,644639	0,32579	0,22789	0,644639	0,410001	0,4657	0,644639	0,629037	0,140351	0,644639	0,32579	0,22789	0,644639	0,410001	0,43557	0,644639	0,629037	0,140352	0,644639	**0,454311775**	11
	4	politics misc	0,314205	0,18561	0,572057	0,235394	0,294534	0,53184	0,141941	0,294534	0,572057	0,339334	0,18561	0,572057	0,28051	0,311296	0,53184	0,179815	0,311296	0,572057	0,339334	0,202679	0,572057	0,28051	0,260085	0,53184	0,179815	0,260085	0,572057	**0,35646112**	16
	5	computer mac hardware	0,617693	0,485014	0,368116	0,617693	0,681641	0,802606	0,617693	0,70257	0,802606	0,708986	0,663874	0,368116	0,708986	0,708986	0,802606	0,708986	0,727339	0,802606	0,708986	0,476033	0,368116	0,708986	0,681641	0,802606	0,708986	0,70257	0,802606	**0,66135742**	2
	6	politics mideast	0,192016	0,239659	0,314869	0,153476	0,209788	0,421431	0,239364	0,209788	0,47598	0,258525	0,220721	0,340565	0,223089	0,215507	0,421431	0,33216	0,215507	0,399832	0,258525	0,236383	0,314869	0,110939	0,215507	0,421431	0,253209	0,215507	0,47598	**0,280965093**	19
	7	computer ibm hardware	0,058597	0,05428	0,609845	0,020328	0,095538	0,726169	0,083288	0,095538	0,453806	0,058598	0,099419	0,609845	0,058597	0,118098	0,726169	0,134708	0,118098	0,453806	0,11907	0,099419	0,609845	0,058598	0,118098	0,726169	0,134708	0,118098	0,453806	**0,259723459**	20
	8	for sale	0,368573	0,488097	0,450034	0,599787	0,429605	0,379015	0,158399	0,429605	0,38691	0,404935	0,792901	0,450034	0,35306	0,324716	0,346698	0,265728	0,324716	0,38691	0,404935	0,792901	0,450034	0,792901	0,324716	0,379015	0,265728	0,324716	0,38691	**0,42450281**	12
	9	science electronics	0,558618	0,558618	0,670174	0,1167	0,558618	0,738929	0,1167	0,511499	0,738929	0,558618	0,558618	0,738929	0,181607	0,558618	0,738929	0,181607	0,558618	0,738929	0,558618	0,558618	0,738929	0,181607	0,558618	0,738929	0,181607	0,558618	0,738929	**0,52584176**	7
	10	computer windows misc	-0,02148	0,361401	0,443303	0,066293	0,546101	0,441178	0,066293	0,44729	0,273379	0,085895	0,320713	0,445499	0,100462	0,546101	0,441178	0,100462	0,546101	0,273379	0,085895	0,357852	0,443303	0,317675	0,546101	0,445499	0,317675	0,546101	0,273379	**0,326556509**	18
	11	motor motorcycle	0,514511	0,376705	0,7445	0,514511	0,157061	0,71691	0,775699	0,285311	0,726304	0,904174	0,381727	0,7445	0,516387	0,158337	0,71691	0,822515	0,294201	0,726304	0,786684	0,410126	0,7445	0,598554	0,158337	0,71691	0,822515	0,294201	0,726304	**0,567951735**	6
	12	sport baseball	0,309385	0,363512	0,767662	0,07559	0,345657	0,939255	0,16471	0,345657	0,844743	0,36075	0,459002	0,767662	0,398629	0,443806	0,939255	0,442542	0,443806	0,844743	0,36075	0,459002	0,767662	0,170357	0,443807	0,939255	0,170357	0,443807	0,844743	**0,513189162**	8
	13	religion christian	0,829337	0,486798	0,763165	0,829337	0,467982	0,795788	0,829337	0,185843	0,800431	0,829337	0,561335	0,763165	0,829337	0,435466	0,829337	0,829337	0,257463	0,800431	0,530564	0,486798	0,763165	0,829337	0,435466	0,795788	0,198785	0,174773	0,800431	**0,634752914**	3
	14	politics guns	0,581668	0,613048	0,705349	0,105805	0,174969	0,557252	0,542526	0,174969	0,656528	0,65205	0,626542	0,707792	0,194646	0,179542	0,557252	0,602407	0,179542	0,656528	0,65205	0,626542	0,705349	0,194646	0,179542	0,587285	0,602407	0,179542	0,656528	**0,476011281**	9
	15	computer graphics	0,168828	0,137318	0,883062	0,168828	0,344492	0,635572	0,738453	0,244713	0,82836	0,14417	0,175244	0,883062	0,14417	0,300067	0,635572	0,738453	0,263687	0,82836	0,263687	0,175244	0,883062	0,263687	0,285741	0,635572	0,738453	0,263687	0,82836	**0,466663195**	10
	16	motor autos	0,251738	0,580364	0,561865	0,251738	0,110328	0,59044	0,251738	0,480771	0,543897	0,280893	0,564897	0,574618	0,280893	0,118752	0,59044	0,280893	0,44884	0,543897	0,280893	0,571424	0,574618	0,280893	0,118752	0,59044	0,280893	0,421837	0,543897	**0,406320247**	13
	17	religion misc	0,333375	0,269041	0,52233	0,639544	0,112325	0,461674	0,41599	0,112325	0,605732	0,293952	0,202025	0,52233	0,639544	0,106376	0,461674	0,5071	0,106376	0,605732	0,191105	0,440328	0,52233	0,639544	0,13103	0,461674	0,41599	0,13103	0,605732	**0,387266967**	14
	18	computer windows x	0,246258	0,582978	0,460816	0,112624	0,378102	0,392636	0,191941	0,275816	0,430332	0,317393	0,678751	0,478909	0,130648	0,302237	0,392636	0,231179	0,319826	0,430332	0,317393	0,678751	0,460816	0,130648	0,318632	0,392636	0,231179	0,319825	0,430332	**0,356800923**	15
	19	science crypt	0,303184	0,185493	0,511226	0,303184	0,163929	0,383692	0,303184	0,185493	0,526853	0,347742	0,20821	0,476407	0,347742	0,186858	0,383692	0,347742	0,20821	0,526853	0,347742	0,172793	0,476407	0,347742	0,186858	0,383692	0,347742	0,172793	0,526853	**0,328233908**	17
AVERAGE SIMILARITY			**0,41367**	**0,385543**	**0,601419**	**0,364591**	**0,338131**	**0,60972**	**0,427925**	**0,328452**	**0,622946**	**0,460793**	**0,434283**	**0,606175**	**0,393737**	**0,34928**	**0,609782**	**0,486276**	**0,352053**	**0,621174**	**0,438833**	**0,434465**	**0,603754**	**0,414648**	**0,327904**	**0,611438**	**0,438492**	**0,334994**	**0,624981**		
SCORE			18	20	9	21	24	6	16	26	2	11	15	7	19	23	5	10	22	3	12	14	8	17	27	4	13	25	1		
NUM SCORE			5	6	3	7	8	2	4	9	1	5	6	3	7	9	2	4	8	1	4	6	3	7	9	2	5	8	1		
NUM AGGREGATED SCORE			14			17			14			14			18			13			13			18			14				
NUM RANKING BY NUM AGGREGATED SCORE			1			3			1			2			3			1			1			3			2				
AGGREGATED SCORE			142									115									121										
QA RANKING BY AGGREGATED SCORE			3									1									2										

Table A.7: Question-Answering models evaluation using "MiniLM-L6-v2" embeddings