information

MDPI

Article

Developing Amaia: A Conversational Agent for Helping Portuguese Entrepreneurs—An Extensive Exploration of Question-Matching Approaches for Portuguese

José Santos [1], **Luís Duarte** [1], **João Ferreira** [1], **Ana Alves** [1,2] **and Hugo Gonçalo Oliveira** [1,*]

1 CISUC, DEI, University of Coimbra, 3030-290 Coimbra, Portugal; santos@student.dei.uc.pt (J.S.); lduarte@student.dei.uc.pt (L.D.); jdcoelho@student.dei.uc.pt (J.F.); ana@dei.uc.pt (A.A.)
2 Instituto Superior de Engenharia de Coimbra (ISEC), Instituto Politécnico de Coimbra, 3030-199 Coimbra, Portugal
* Correspondence: hroliv@dei.uc.pt

Received: 31 July 2020; Accepted: 28 August 2020; Published: 1 September 2020

Abstract: This paper describes how we tackled the development of Amaia, a conversational agent for Portuguese entrepreneurs. After introducing the domain corpus used as Amaia's Knowledge Base (KB), we make an extensive comparison of approaches for automatically matching user requests with Frequently Asked Questions (FAQs) in the KB, covering Information Retrieval (IR), approaches based on static and contextual word embeddings, and a model of Semantic Textual Similarity (STS) trained for Portuguese, which achieved the best performance. We further describe how we decreased the model's complexity and improved scalability, with minimal impact on performance. In the end, Amaia combines an IR library and an STS model with reduced features. Towards a more human-like behavior, Amaia can also answer out-of-domain questions, based on a second corpus integrated in the KB. Such interactions are identified with a text classifier, also described in the paper.

Keywords: semantic textual similarity; question answering; conversational agents; machine learning; information retrieval; text classification

---

1. Introduction

ePortugal (https://eportugal.gov.pt/) is a web portal managed by the Portuguese Administrative Modernization Agency (AMA), which "aims to facilitate the interactions between citizens, companies and the Portuguese State, making them clearer and simpler". Among others, it provides information on public administration services, which may, indirectly, answer a broad range of questions. However, due to the huge amount of information on different services, involving significantly different procedures, and thus also organized differently, some answers can be hard to get or take too much time to find.

In order to make the process of finding answers for entrepreneurs easier, about two years ago, we were challenged to develop an alternative interface to *Balcão do Empreendedor* (BDE, in English, Entrepreneur's Desk), now incorporated in ePortugal. Beyond a search interface, it would enable interested users to make questions, in natural language, to be answered automatically, also in natural language, thus avoiding to explore the site, and spending time on navigation and reading long documents. In fact, the challenge was to develop a computational agent that, among other conversational skills, would be apt to help entrepreneur's willing to develop an economic activity in Portugal, by providing answers to their questions.

Given the limitations of end-to-end conversational agents, and once we noticed that lists of Frequently Asked Questions (FAQs) were available for some of the target services, we decided to develop a retrieval-based agent. This option would also allow us to focus on two components of the work, independently: (i) the "Knowledge Base" (KB), which would contain all the questions that the agent would be able to answer, as well as their answers; (ii) retrieval, which would encompass all the processing required to the user input and how it would be exploited for searching for and retrieving a suitable answer from the KB. This approach also had in mind that both adding questions or adapting the agent to a different domain would be mostly a matter of changing the KB.

The KB would consist of FAQs from BDE and ePortugal that, along the duration of this project, were provided by AMA. However, the main focus was on the retrieval component. In order to select the approach to follow, different approaches and technologies were explored for matching user requests with FAQs in the KB and providing their answers. To go beyond traditional Information Retrieval (IR) [1], we looked at Semantic Textual Similarity (STS, [2,3]) as a useful task to tackle for this process, i.e., user requests would be matched to the most semantically-similar questions in the KB.

STS aims at computing the proximity of meaning of two fragments of text. Shared tasks on the topic have been organized in the scope of SemEval 2012 [2] to 2017 [3], targeting English, Arabic, and Spanish. For Portuguese, there have been two editions of a shared task on the topic, dubbed ASSIN [4,5]. In this work, we explore different approaches for STS, including simple unsupervised methods, based on IR, word overlap, or on pre-trained models of word embeddings. We further exploit several features and train our own STS model in the ASSIN collections. Such approaches are tested in a corpus created for this purpose, AIA-BDE [6], which, besides the FAQs in the KB, contains their variations, simulating user requests.

Following an extensive comparison of the previous approaches and the discussion of their results, we came to the conclusion that the supervised STS model is a good option. However, since it relies on many features, we also: (i) reduce its complexity through feature selection; and (ii) combine it with an IR library for a preliminary selection of candidate questions. The previous options were supported by a set of experiments, which confirmed that the impact on question-matching performance was minimal.

Moreover, we aimed at developing an agent that would not simply answer domain questions, but with which it would be possible to have a lighter conversation, more or less on any topic, or at least simulate this capability. For this purpose, we compiled another corpus, this time for Out-Of-Domain (OOD) questions and answers, i.e., chitchat. Some of those questions were added manually, while others came from a corpus of movie subtitles [7].

The resulting agent was dubbed Amaia, and we see it as an evolution of Cobaia, described in our previous paper [8], for which this is an extended version. The main differences of the present work are the following:

- We compare the supervised STS model with a broader range of unsupervised approaches for STS and make a more thorough selection of features, also considering the complexity of the model;
- Amaia uses and is assessed in a new version of the AIA-BDE corpus;
- Amaia relies on a more flexible strategy for identifying OOD interactions, based on a classifier, and provides answers to such questions based on a smaller and more controlled corpus.

The remainder of the paper is organized as follows: Section 2 overviews related work on conversational agents and IR-based natural language interfaces to FAQs; Section 3 describes the corpora used in this work, namely the AIA-BDE corpus, used both as Amaia's KB and for evaluation purposes, and the Chitchat corpus, which Amaia resorts to for handling OOD interactions; Section 4 discusses the performance of several unsupervised approaches for STS in the AIA-BDE corpus; Section 5 describes how a model can be trained for STS in Portuguese and then applied to AIA-BDE, also including a discussion on the selection of the most relevant features; Section 6 is on the approach for dealing with OOD interactions, which includes training a classifier for discriminating between such interactions and domain questions. Before concluding in Section 8, Section 7 wraps everything with the integration of the IR library, the STS model, and the OOD classifier, as well as the created corpora, in Amaia, illustrated with an example of a conversation.

2. Related Work

Dialogue systems typically exploit large collections of text, often including conversations. End-to-end generative systems model conversations with a neural network that learns to decode sequences of text (e.g., interactions) and translate them to other sequences (e.g., responses) [9]. Such systems are generally scalable and versatile, always generate a response, but have limitations for performing specific tasks. As they make few assumptions on the domain and generally have no access to external sources of knowledge, they can rarely handle factual content. They also tend to be too repetitive and provide inconsistent, trivial or meaningless answers.

Domain-oriented dialogue systems tend to follow other strategies and integrate Information Retrieval (IR) and Question Answering (QA) techniques to find the most relevant response for natural language requests. In traditional IR [1], a query represents an information need, typically in the form of keywords, to be answered with a list of documents. Relevant documents are generally selected because they mention the keywords, or are about the topics they convey. Automatic QA [10], diversely, finds answers to natural language questions. Answers can be retrieved from a structured KB [11] or from a collection of documents [12]. This has similarities to IR, but queries have to be further interpreted, possibly reasoned—where Natural Language Understanding (NLU) capabilities may be necessary—while answers are expected to go beyond a mere list of documents.

Given a user input, IR-based conversational agents search for the most similar request on the corpus and output their response (e.g., [13]). They rely on an IR system for efficiently indexing the documents of the corpus and, in order to identify similar texts and computing their relevance, a common approach is to rely on the cosine between vector representations of the query and of the indexed texts, where words can be weighted according to their relevance, with techniques such as TF-IDF. Instead of relying exclusively on the cosine, an alternative function can be learned specifically for computing the relevance or relatedness of a document for a query. This can be achieved, for instance, with a regression model that considers several lexical or semantic features to measure Semantic Textual Similarity (STS, [14]). This is also a common approach of systems participating in STS shared tasks (e.g., [3]), some of which covers pairs of questions and their similarity [15]. A related shared task is Community Question Answering [16,17], where similarity between questions and comments or other questions is computed, for ranking purposes.

STS can also be useful in the development of natural language interfaces for lists of FAQs. Due to their nature and structure, the latter should be seen as valuable resources for exploitation. On this context, there has been interest in SMS-based interfaces for FAQs [18], work for QA from FAQs in Croatian [19], and a shared task on this topic, in Italian [20]. FAQ-based QA agents often pre-process text in questions, answers, and user requests, applying tokenization and stopword removal operations. For retrieving suitable answers, the similarity between user queries and available FAQs is computed by exploiting word overlap [19], the presence of synonyms [18,21], or distributional semantic features [19,22].

In opposition to generative systems, IR-based dialogue systems do not handle very well requests for which there is no similar text in the corpus. However, an alternative IR-based strategy can still be followed, in this case, for finding similar texts in a more general corpus, such as movie subtitles [23]. Either with an IR or generative approach, an important challenge is to give consistent responses. For this purpose, there are different approaches for developing conversational agents with a persona. In the generative domain, persona embeddings can be incorporated [24], while in the IR-domain, this issue has been tackled by including a smaller corpus of personal questions and answers [25].

3. Corpora

Our main goal was to develop a conversational agent that would answer questions related to entrepreneurship and performing economic activity in Portugal. To some extent, it would be an alternative channel to searching for the necessary information in the former *Balcão do*

Empreendedor (BDE, in English, Entrepreneur's Desk) and related services, now incorporated in the e-Portugal website.

However, towards a more human behavior, we also wanted the agent to enable basic open-domain conversations. Therefore, we have also compiled the Chitchat corpus, to be used instead of AIA-BDE for answering out-of-domain (OOD) interactions. This section describes both corpora and, later, Section 6 explains how Amaia discriminates between domain and OOD interactions.

3.1. The AIA-BDE Corpus

In order to create what we later called the agent's KB, we asked AMA for data available in BDE that would be valuable to such an agent and, at the same time, easy to integrate and exploit. Once we found that several services had FAQs, we decided to focus the development of the KB around them, and get some inspiration from related work on FAQ-based agents [18–20]. At the same time, we aimed to develop an agent with a flexible architecture that would be easily adapted to other domains, and this also seemed like a good option from that perspective.

FAQs were collected and compiled in the KB, so that the agent would access questions and their answers. The agent would thus try to match natural language user requests with questions for which it had an answer, i.e., they were answered by a FAQ in the KB. Once it identifies the most similar FAQ to the request, it may provide its answer to the user.

We baptized the corpus of FAQs as AIA-BDE, and it is now in the second version, the one used in this paper, with more questions and more variations than its first version [6]. More precisely, it contains 855 FAQs from four different sources: *Espaço Empresa* (EE, Business Spot, 625 FAQs), *Apoios Sociais* (AS, Social Support, 56 FAQs), *Regime de Acesso a Atividades de Comércio, Serviços e Restauração* (RJACSR, Access Regime to Commerce, Services and Catering Activities, 118 FAQs), and *Alojamento Local* (AL, Local Accommodation, 56 FAQs) (For those interested, both versions of the AIA-BDE corpus are available from https://github.com/NLP-CISUC/AIA-BDE).

However, in addition to the FAQs, AIA-BDE also contains their variations, which are paraphrases or related questions using other words, sometimes omitting information. To some extent, such variations simulate user requests for which answers are available. Therefore, we may use them for assessing our agents on this domain, i.e., how well they can match variations with the original questions.

There are at least five variations for each question, and some have up to 12. Though, as there is no perfect way of creating variations, and because their manual creation is time-consuming, variations were produced along the time, following significantly different approaches, and by different people. Therefore, since matching variations created differently may pose different challenges, we marked variations according their creation process, namely:

- Two types generated automatically, with the help of the Google Translate API (https://cloud.google.com/translate/docs/), as follows: translation of the Portuguese question to English and back to Portuguese (VG1); the previous result back to English and back to Portuguese (VG2). A variation of each type is available for every original question in the corpus. Most of them end up being very similar to the original questions, and several VG2 are the same as the corresponding VG1. In addition, a minority of these variations has errors or imprecisions resulting from the translation process.
- Three types of manually created variations. Two were created by different controlled groups of native Portuguese speaking volunteers, one consisting of the authors of this paper (VUC), and another produced by Computer Science undergraduates of the Technical University of Lisbon (VIN). A third type was the result of a crowdsourcing task in the Amazon Mechanical Turk (https://www.mturk.com/) platform (VMT). For all the types, volunteers were instructed to read the original question and answer, and to rewrite the question using other words, but keeping its original meaning or, in some cases, a meaning entailed by it.

However, not all FAQs have all types of variation. VMT variations are only available for the AS FAQs, with each having three of them. VIN variations are only available for the EE FAQs, but each of those FAQs may have from 0 to 9 of such variations, all different. Finally, there are VUC variations for most of the EE FAQs, as well as for all the RJACSR and AL, each with between 2 and 10 of such variations.

Table 1 illustrates the AIA-BDE corpus, with examples of a question from each source, some of its variations, and its answer. When they are the same as the VG1, VG2 variations are omitted.

Table 1. Examples of the AIA-BDE corpus.

Source	Var	Text
EE	P	*Qual o custo de constituição de uma "Empresa na Hora"?* (What is the cost of setting up a "Company on the Spot"?)
	VG1	*Qual é o custo de configurar um "Negócio no local"?* (What is the cost of setting up a "Business on site"?)
	VUC	*Preço para constituir uma empresa na hora.* (Price for setting up a company on the spot.)
	VIN	*Quanto terei de pagar para ter uma empresa na hora?* (How much will I have to pay to have a company on the spot?)
	R	*O custo de constituição de uma sociedade é de €360, incluindo publicações ...* (The cost of setting up a company is €360, including publications ...)
AS	P	*Quando é que me dão uma resposta sobre o apoio social a crianças e jovens?* (When can you give me an answer on social support for children and young people?)
	VG1	*Quando recebo uma resposta sobre apoio social para crianças e jovens?* (When do I get a response on social support for children and youth?)
	VG2	*Quando recebo uma resposta sobre apoio social a crianças e jovens?* (When do I get a response on social support for children and youth?)
	VMT	*Tenho que esperar muito para ter uma resposta sobre o apoio social a crianças e jovens?* (Do I have to wait too long to get an answer on social support for children and young people?)
	R	*Depois de fazer a sua inscrição na instituição que lhe interessa, pode acontecer ter de ficar em lista de espera...* (After registering at the institution you are interested in, you may have to wait on the waiting list...)
RJACSR	P	*Qual a coima aplicável às contraordenações graves?* (What is the fine for serious offences?)
	VG1	*Qual é a multa aplicável à falta grave?* (What is the fine applicable to serious misconduct?)
	VUC	*coima para contraordenação grave* (fine for serious offence)
	R	*As contraordenações graves são sancionáveis com coima: ...* (Serious offences are punishable with a fine ...)
AL	P	*No alojamento local é obrigatória a certificação energética? Em que termos deve ser efetuada?* (Is energy certification compulsory in local accommodation? In what terms should it be done?)
	VG1	*No alojamento local é obrigatório a certificação energética? Em que condições deveria ser feito?* (Is energy certification compulsory in local accommodation? Under what conditions should it be done?)
	VG2	*A certificação energética é necessária em alojamento local? Em que condições deve ser feito?* (Is energy certification required in local accommodation? Under what conditions should it be done?)
	VUC	*Como deve ser feita certificação energética do meu alojamento local?* (How should the energy certification of my local accomodation be done?)
	R	*De acordo com esclarecimento da DGEG (Direção-Geral de Energia e Geologia) ...* (According to DGEG (General-Direction of Energy and Geology) ...)

3.2. The Chitchat Corpus

The Chitchat corpus has the same format as AIA-BDE, but includes OOD interactions and its responses, acquired from two different sources:

- Handcrafted set of 22 personal questions, i.e., questions that are commonly made in chats, and their responses;
- About 1500 interaction–response pairs obtained from the Portuguese part of the Subtle [7] corpus of movie subtitles.

Subtitles are indeed a great source of material for chitchat. However, we soon noticed that, when using too many subtitles with no selection criteria, conversations could easily become impractical. Therefore, we selected a subset of interactions that occur 50 or more times in Subtle, as well as their most frequent response (For those interested, the Chitchat corpus is available from https://github.com/NLP-CISUC/AIA-BDE). In this process, interactions and responses with strange characters and proper nouns were ignored.

Table 2 illustrates the Chitchat corpus with some examples of its entries.

Table 2. Examples of the Chitchat corpus.

Source	Type	Text
Personal	P	*Quem és tu?* (Who are you?)
	R	*Olá, eu sou a Amaia, um sistema inteligente para apoio a empreendedores, desenvolvido em 2019 no Centro de Informática e Sistemas da Universidade de Coimbra.* (Hello, I'm Amaia, an intelligent system for supporting entrepreneurs, developed in 2019 at the Center for Informatics and Systems of the University of Coimbra.)
	P	*Que idade tens?* (How old are you?)
	R	*Essa pergunta não se faz a uma senhora, mas será que isso importa assim tanto? Em que o posso ajudar?* (This question should not be asked to a lady, but does it really matter that much? How can I help you?)
	P	*És um chatbot?* (Are you a chatbot?)
	R	*Sou um sistema inteligente de resposta automática a perguntas, mas podes tentar conversar comigo.* (I'm an intelligent system for answering questions, but you can try chatting with me.)
Subtitles	P	*Até amanhã.* (See you tomorrow.)
	R	*Adeus.* (Goodbye.)
	P	*Prazer em conhecê-lo.* (Nice to meet you.)
	R	*Igualmente.* (Me too.)
	P	*Que fazes aqui?* (What are you doing here?)
	R	*E tu?* (What about you?)
	P	*O que queres dizer com isso?* (What do you mean?)
	R	*Nada.* (Nothing.)

4. Answering AIA-BDE with Unsupervised Approaches

As the AIA-BDE corpus allows for the assessment of different approaches when matching variations (i.e., simulations of user requests) with actual questions, we used it as a benchmark in this task. In this section, we look at the performance of several unsupervised approaches for STS, in the sense that they rely exclusively on the existing data and, in some cases, on pre-trained embeddings. The first is a traditional IR approach, based on a full text search library, used for indexing and searching the text, according to different configurable parameters. The second group of approaches is based on vector representations of text, which can be created directly from the data, or based on pre-trained models of word embeddings. To some extent, these approaches could be seen as baselines. However, as we show throughout the paper, some rely on very powerful language models that lead to high performances.

In both cases, performance was measured by computing the accuracy of each approach in all the variations of the AIA-BDE corpus. Moreover, having in mind that, in many scenarios, it is better to return a smaller set of answers that include the correct one, than to give no answer or return one that is incorrect, accuracy was also measured for the presence of the correct answer in the top-3 or top-5 best-ranked candidates.

4.1. Traditional IR

For testing a traditional IR approach, we relied on Whoosh (https://whoosh.readthedocs.io), a Python full text search library, which builds an index for a corpus and enables efficient text-based searches on it. More precisely, Whoosh was used for indexing the AIA-BDE corpus, such that each FAQ was represented by two fields, the question and the answer, with searches made on the question. Despite using the same corpus, Whoosh provides different ranking functions and analyzers that may be used, some of which for Portuguese. We tested both BM25F and Frequency scoring functions, opting for the former due to the poor performance of the latter, whose accuracy remained below 15% on all our tests. The *group* parameter value of the query parser was changed to *OrGroup*, which makes the terms in the query optional. When compared to the default setting (see our previous paper [8]), this improves the matching performance significantly.

In addition to the default indexation, Whoosh also allows for the application of a set of filters, possibly included in an analyzer, which may differ in how text is tokenized, or how tokens are normalized. In this work, the following configurations were compared:

- **Default** Whoosh configuration.
- **Default + Fuzzy**, the default configuration with Fuzzy Search, which enables partial matches (e.g., spelling mistakes).
- **LanguageAnalyzer**, which converts words to lower-case, removes Portuguese stopwords, and converts words to their stem, following Portuguese rules.
- **Stemming Analyzer**, a simplification of the previous that does not remove stopwords.
- **Stemming Analyzer + Charset Filter**, the Stemming Analyzer followed by a filter that removes graphical accents.
- **N-gram Filter (2–3)**, which tokenizes text and indexes it according to character n-grams of sizes 2 and 3.
- **N-gram Filter (2–4)**, which tokenizes text and indexes it according to character n-grams of sizes 2, 3 and 4.

Table 3 shows the accuracy of the previous configurations when matching the original questions with themselves, for sanity check, and with the set of all available variations in AIA-BDE. Since Whoosh may return more than a single result, i.e., a ranked list with the most relevant results, we can also look for the presence of the correct question in the top-*n* results. Thus, in addition to the first in the rank (Top1), the table presents the proportion of questions for which the correct match was in the top-3 and top-5. This has also in mind that, even in a real application scenario, missing the correct match might be minimized by presenting the top-*n* matches, hoping that one of them will be correct.

As expected, the great majority of questions is correctly matched with itself, which shows that the traditional IR approach is doing its job well. The minority of questions not matched are short questions that share the majority of tokens with others. For instance, with the Default configuration, this includes mostly questions with a single different word, such as: *O que é um certificado digital?* and *O que é um certificado digital qualificado?*, or *Quem é o franchisador?* and *Quem é o franchisado?*. As for the variations, the Stemming Analyzer leads to the best results, especially with the Charset Filter. We recall that the only difference between the Language and the Stemming Analyzer is that the latter does not remove stopwords, which shows that, in opposition to other tasks, stopwords are important here. With the best configuration, the proportion of correct matches is close to 80%, with almost 90% in the top-3 and more than 92% in the top-5. This confirms that, given a user request, considering more than a single question may significantly increase the chance of giving the right answer. Table 4 shows the accuracy for the variations of each type.

Table 3. Proportion of original questions and variations correctly matched with original FAQs (Top1), in Top-3 and in Top-5 most similar, using different configurations of Whoosh.

Configuration	Original (855)			Variations (4973)		
	Top1	Top3	Top5	Top1	Top3	Top5
Default	96.4	99.2	99.4	72.3	83.3	86.0
Default Fuzzy	93.3	97.9	98.5	75.5	85.8	88.4
Language	93.7	97.9	98.5	75.4	85.8	88.4
Stem	**98.3**	**100.0**	**100.0**	79.4	89.2	91.6
Stem + Charset	94.2	98.0	98.5	**79.7**	**89.7**	**92.5**
Ngram (2-3)	97.4	99.9	99.9	6.4	22.2	29.0
Ngram (2-4)	97.8	99.8	99.8	46.8	70.9	76.9

Table 4. Proportion of variations of different types correctly matched with original FAQs (Top1), in Top-3 and in Top-5 most similar, using different configurations of Whoosh.

Configuration	Variation														
	VG1 (855)			VG2 (855)			VIN (2279)			VUC (816)			VMT (168)		
	Top1	Top3	Top5	Top1	Top3	Top5	Top1	Top3	Top5	Top1	Top3	Top5	Top1	Top3	Top5
Default	83.2	90.9	93.1	80.2	88.3	90.5	73.8	85.7	88.2	50.9	65.6	69.5	59.5	73.8	79.2
Default+Fuzzy	86.0	94.3	95.2	83.7	92.6	93.8	76.9	87.2	89.7	55.2	68.1	72.7	58.9	**75.6**	83.3
Language	82.3	90.6	93.0	79.0	88.3	91.0	80.1	90.2	92.8	55.2	73.0	78.1	52.4	**75.6**	**86.3**
Stem	86.2	94.2	95.4	84.4	92.3	94.2	76.4	87.4	90.2	54.7	68.1	71.6	**60.7**	74.4	81.0
Stem+Charset	**88.1**	**95.6**	**96.6**	**85.4**	**93.7**	**95.6**	**82.3**	**91.4**	**94.0**	**62.5**	**78.9**	**83.0**	54.8	70.2	81.5
Ngram (2-3)	7.8	28.3	36.1	6.7	26.5	33.5	7.0	21.7	28.6	3.9	15.6	21.0	3.0	9.5	13.7
Ngram (2-4)	55.3	80.5	85.6	52.4	77.1	82.5	48.1	71.2	77.1	30.8	56.1	63.6	35.1	59.5	66.7

As expected, the highest performance is for the VG1 and VG2 variations because, in terms of surface text, they are closer to the original questions. Nevertheless, accuracy is significantly lower than for the original questions (about 10 points for the top-1, and 3 for the top-5, considering the best configuration in both). Manually-created variations are the most difficult to match correctly, especially VUC and VMT. With the best configuration, about 62% of the VUC variations is matched correctly, and 83% in the top-5. VMT variations are also those for which the best performance, 60% correct matches, is achieved without the Charset Filter, and for which the best performance for the top-3 (75%) and top-5 (86%) is achieved with the Language Analyzer. Nevertheless, from these figures, we would decide to use Whoosh with the Stemming Analyzer and the Charset Filter. With the N-gram filter, performance decreases significantly for all variations, especially when 4-grams are not included, so it would not be a viable option.

4.2. Word Vector Approaches

In the second group of approaches, each sentence was represented by a fixed-length vector of numbers, and similarity was computed with the cosine between the vector representation of each variation and the vector representation of all original questions. Different methods were used for representing the sentence as a vector, including traditional approaches, where the vector representation considers only the vocabulary of our data and the surface text, but also approaches based on pre-trained models of static word embeddings and state-of-the-art contextual embeddings. The traditional methods tested were based in the following scikit-learn [26] implementations:

- **Count Vectorizer**, which converts each sentence to a vector of token counts.

- **TFIDF Vectorizer**, which converts each sentence to a vector of TF-IDF features, i.e., the weight of each token increases proportionally to count, but is inversely proportional to its frequency in the corpus, in this case, the original questions of AIA-BDE.

Both were used with default parameters, meaning that sentences were represented by sparse vectors with a fixed-size equal to the size of the vocabulary.

In approaches based on static word embeddings, the sentence vector is computed from the vector of each of its words, according to a pre-trained model. In this process, tokens without alpha-numeric characters (e.g., punctuation signs) and tokens not covered by the model are ignored. Moreover, all words may have the same weight, resulting in the average embedding, or they can be weighted by the relevance of each word, given by the **TF-IDF**, again computed in the original questions. Four different pre-trained models of this kind were tested in this experiment, learned with the following algorithms:

- **word2vec** [27], namely its two common variations of **CBOW** and **SKIP-GRAM**;
- **GloVe** [28], a common alternative to word2vec;
- **FastText** [29], as an attempt to better deal with the Portuguese morphology, given that it considers character n-grams.

The word2vec and GloVe models pre-trained for Portuguese were obtained from the NILC word embeddings repository [30]. For FastText, we used a different source, trained by the creators of this algorithm (https://fasttext.cc/). All of them had vectors with 300 dimensions and were loaded with the Gensim Python library [31].

Approaches based on contextual embeddings relied on **BERT** [32], a recent model that encodes words and longer sequences based on a Transformer neural network. In this case, full sentences were encoded directly by BERT, which resulted in their vector representation. Two pre-trained BERT models were used for this purpose:

- BERT-Base, Multilingual Cased (Multilingual BERT), trained by the creators of BERT (https://github.com/google-research/bert) for 104 languages, which encodes given text in 768-sized vectors;
- bert-large-portuguese-cased (Portuguese BERT) [33], trained specifically for Portuguese, which encodes given text in 1024-sized vectors.

BERT models were loaded with the bert-as-a-service tool (https://github.com/hanxiao/bert-as-service), with default options, except for the maximum length of sequences, set to NONE for dynamically using the longest sequence in the batch.

Similarly to Table 3, Table 5 shows the accuracy of the approaches based on the previous models when matching the original questions, for sanity check, and for the set of all variations in AIA-BDE. As in the previous section, accuracy is obtained from the number of variations for which the correct question was the most similar. For the top-3 and top-5, the correct question must be in the top-3 and top-5 most similar, respectively.

The first observation is that word vectors that are learned from external sources of text lead to better performances than the Count and the TF-IDF, which are computed from the questions of AIA-BDE and rely only on the surface text. Another observation is that, with the pre-trained word embeddings, performance decreases with TF-IDF. Although TF-IDF would give more weight to more relevant words, this is only based on the questions of AIA-BDE, which are probably not enough for computing proper weights. Another reason for this may be related to the role of stopwords. TF-IDF should give them less weight, but the previous experiments with Whoosh suggested that removing stopwords had a negative impact on performance.

Table 5. Proportion of original questions and variations correctly matched with original FAQs (Top1), in Top-3 and in Top-5 most similar, using different Word Vector approaches.

Approach	**Original (855)**			**Variations (4973)**		
	Top1	**Top3**	**Top5**	**Top1**	**Top3**	**Top5**
CountVectorizer	32.3	**100.0**	**100.0**	4.0	23.3	24.5
TFIDF-Vectorizer	98.8	**100.0**	**100.0**	4.0	4.7	5.5
CBOW	**99.0**	**100.0**	**100.0**	**79.8**	**89.8**	**92.1**
CBOW + TF-IDF	98.8	**100.0**	**100.0**	61.5	76.0	80.4
SKIP	**99.0**	**100.0**	**100.0**	78.8	88.3	91.0
SKIP + TF-IDF	98.8	**100.0**	**100.0**	57.5	71.1	76.0
FastText	**99.0**	**100.0**	**100.0**	41.8	52.7	57.3
FastText + TF-IDF	98.8	**100.0**	**100.0**	33.2	41.8	46.0
GloVe	**99.0**	**100.0**	**100.0**	70.6	79.9	82.8
GloVe + TF-IDF	98.8	**100.0**	**100.0**	43.7	55.3	59.7
Multilingual BERT	98.9	**100.0**	**100.0**	73.9	82.9	86.0
Portuguese BERT	98.9	**100.0**	**100.0**	79.0	88.9	91.1

Different performances are achieved by different models, with the best achieved by the word2vec-CBOW, without TF-IDF. Its figures are comparable to the best achieved with Whoosh. Surprisingly, none of the state-of-the-art BERT models could outperform word2vec. Out of the two, the best was the Portuguese BERT, which makes sense because it was trained exclusively for Portuguese. Its performance is comparable to word2vec-SKIP, which is the second best model. Table 6 shows the accuracy for the variations of each type.

Table 6. Proportion of variations of different types correctly matched with original FAQs (Top1), in Top-3 and in Top-5 most similar, using different Word Vector approaches.

Approach	**Variation**														
	VG1 (855)			**VG2 (855)**			**VIN (2,279)**			**VUC (816)**			**VMT (168)**		
	Top1	**Top3**	**Top5**	**Top1**	**Top3**	**Top5**	**Top1**	**Top3**	**Top5**	**Top1**	**Top3**	**Top5**	**Top1**	**Top3**	**Top5**
CountVectorizer	6.1	40.0	41.8	5.5	35.2	36.7	3.3	17.4	18.2	1.4	9.4	9.4	7.7	26.2	33.3
TFIDF-Vectorizer	9.1	9.7	10.3	7.5	7.8	8.4	2.0	2.7	3.6	0.6	0.9	1.1	3.0	9.5	15.5
CBOW	88.4	**95.8**	**97.1**	86.5	94.7	96.4	82.2	91.0	93.3	59.1	**75.5**	**79.8**	**70.8**	**86.3**	**89.9**
CBOW + TF-IDF	71.5	84.8	87.8	67.8	81.8	86.0	64.3	78.3	82.4	38.7	53.8	60.4	50.6	76.8	83.3
SKIP	88.9	95.7	96.6	87.1	94.5	95.8	80.4	89.2	91.7	57.2	72.3	78.4	69.6	83.9	89.9
SKIP + TF-IDF	67.0	80.8	84.4	64.4	76.4	80.9	61.1	74.1	78.5	32.6	47.1	53.6	46.4	72.6	81.0
FastText	57.2	68.5	72.6	51.5	63.4	68.0	40.0	51.0	55.2	21.0	28.8	34.2	39.9	57.7	64.9
FastText + TF-IDF	43.9	48.7	53.3	40.2	48.7	53.2	32.6	42.5	46.2	15.8	23.8	27.9	36.3	50.0	56.5
GloVe	85.1	92.0	93.9	82.5	90.3	92.2	70.0	79.2	81.7	46.3	58.0	64.6	63.7	79.8	83.3
GloVe + TF-IDF	49.1	60.2	63.7	45.8	56.6	60.6	48.9	60.9	65.1	21.4	32.4	37.5	42.3	61.3	70.2
Multiling BERT	**90.6**	95.3	96.6	**90.6**	**96.0**	**97.5**	73.7	84.0	87.1	46.4	59.6	65.8	39.9	53.0	56.5
Portuguese BERT	86.1	94.9	96.1	83.6	93.1	94.5	**83.5**	**92.4**	**94.1**	**60.4**	75.2	79.4	47.6	57.1	62.5

Again, performance is higher for VG1 and VG2 and lower for the manually-created variations. However, these figures show that the selection of the best model is not as straightforward as it was for Whoosh, with different models having the best performance for different variations. For instance, Multilingual BERT achieved the best performance for VG1, considering only the first result, and VG2, possibly because these variations are generated with the help of machine translation and this BERT model is multilingual. Moreover, since this model was trained by Google, it is also possible that it is somehow used by Google Translate. The best performance in the VIN variations (83.5%, about 1 point higher than the best with Whoosh) is by the Portuguese BERT, the same model that achieves the best performance in the VUC (60.4%, about 2 points lower than the best with Whoosh). However, this happened only for the first result, with word2vec-CBOW slightly improving in the top-3 and top-5. This was also the best model for the VMT variations and, when considering the top-3 and top-5 in VG1, which is why it was the best model overall.

Considering also that word2vec is less complex than BERT, out of the tested models, it would be our choice. However, we believed that these figures could be further improved if several models were combined, and possibly combined with other features. Therefore, in the next section, we describe how different features can be exploited for learning a model of Portuguese STS that suits our purpose.

5. A Model for Portuguese STS

After testing several unsupervised approaches, which, due to their easy implementation, can be seen as baselines, we leveraged on available data for training an STS model for Portuguese, which would hopefully improve the performance of the baselines. The goal was to develop a model as broad as possible that would exploit many potentially useful features. However, at the same time, we did not want it to become too complex, which is why we tried to use only a fraction of all the features that we could extract. The development of the STS model followed a supervised learning approach. It was validated, trained, and tested in sentence pairs from the collections of ASSIN [4] and ASSIN 2 [5], which comprise a total of ≈20,000 pairs with annotated similarity scores, based on human opinions, ranging between 1 (completely different) and 5 (equivalent).

Before concluding the section, we describe how the IR approach can be combined with the STS model for reducing the number of necessary computations.

5.1. Training a Model for Portuguese STS

To compute the STS between sentence pairs, a broad set of 64 features was initially extracted, covering different types of features, namely lexical, syntactic, semantic, and distributional. Features were extracted with the help of the following Python libraries: NLTK [34], for getting token and character n-grams; NLPyPort [35] (i.e., NLTK with some improvements for Portuguese), for getting Part-of-Speech (PoS) tags, named entities and lemmas; Gensim [31] and scikit-learn [26], for extracting distributional features, which included the word embeddings in Section 4.2 and others.

However, as mentioned earlier, we wanted to avoid a very complex model. Therefore, even before training and testing any model, we tried to reduce the dimensionality of the feature set. For this purpose, we ran Recursive Feature Elimination (RFE), available in scikit-learn, to select the most relevant features out of the initial 64. This method requires an external estimator for assigning weights to features according to their respective importance. In this case, we chose a Random Forest Regressor (RFR) model as the estimator. Even though other algorithms could have been used for this purpose, we had previous experience with the RFR in a similar context. Starting with the initial set of features, the estimator is repeatedly trained until the desired number of features is reached, by removing the least important feature from the group at each iteration. We tested different thresholds for the number of features to be selected, ranging from 20% (top-13 features) to 80% (top-51 features) of the original set, and evaluated the performance of each test with the coefficient of determination R^2 of the prediction, which allowed us to select the threshold value. To avoid overfitting, this process was run in a validation set comprising of 10% of the sentence pairs in the ASSIN and ASSIN-2 training collections, selected randomly.

The best performance was achieved with a threshold of 42%, meaning that the initial set of 64 features could be reduced to 27. This includes the following features:

- Jaccard coefficient computed between the sets of token 1-grams (1).
- Overlap and Dice coefficients, each computed between the sets of token 1/2-grams (4).
- Jaccard, Overlap, and Dice coefficients, each computed between the sets of character 2/3/4-grams (9).
- Cosine of averaged token vectors, i.e., as in Section 4.2, computed with the following word embeddings: NILC word2vec-CBOW (300-sized), NILC GloVe (300-sized), fastText.cc [29], Numberbatch [36], and PT-LKB [37] (5).
- Cosine of TF-IDF-weighted averaged token vectors, computed with the previous word embeddings (5).
- Cosine of TF-IDF-weighted token vectors (1).
- Absolute difference of used adverbs, computed with the NLPyPort's PoS tagger (1).

- Jaccard coefficient computed between the triples of syntactic dependency relations, obtained with *spaCy*'s (https://spacy.io) dependency parser (1).

These features are also summarized in Table 7. We note that most models of static word embeddings tested in Section 4.2 were in the set of 64. BERT contextual embeddings, on the other hand, were not included, due to the large memory requirements of these models. What is curious to see is that, even though fastText was one of worst-performing methods back then, it was selected, possibly due to its complementary nature.

Table 7. Features in the reduced set.

Lexical Features (15)		
Metric	**Tokens**	**Characters**
Jaccard	1-grams	2-grams, 3-grams, 4-grams
Overlap	1-grams, 2-grams	2-grams, 3-grams, 4-grams
Dice	1-grams, 2-grams	2-grams, 3-grams, 4-grams
TF-IDF	1-grams	
Syntactic features (2)		
Metric	**Description**	
Jaccard	Triples of syntactic dependencies	
Difference	Adverbs	
Distributional Semantic features (10)		
Cosine(token vectors)	**Models**	
Average	word2vec-CBOW, GloVe, fastText.cc, Numberbatch, PT-LKB	
TF-IDF weighted	word2vec-CBOW, GloVe, fastText.cc, Numberbatch, PT-LKB	

With this feature set, we explored different regression algorithms available in scikit-learn. They were trained in the remaining training pairs of ASSIN and ASSIN-2 (90%, after removing the 10% used for feature selection), and tested in each of the three test collections available. In those experiments, a Support Vector Regressor (SVR) and a Random Forest Regressor (RFR), both using default parameters, stood out, with comparable results. However, we decided to stick with the SVR because we had already used it in our previous work, with both ASSIN [38] and AIA-BDE [8].

5.2. Further Reducing the Size of the Model

Even though we were able to reduce the feature set considerably with RFE, we had an intuition that this set could be further reduced. Our intuition mainly relied on the fact that the reduced set includes five models of word embeddings that, although learned with different algorithms, should be somehow overlapping. Moreover, we would like to analyze whether we could get rid of two features that require the use of two external libraries, namely the syntactic dependencies, which require spaCy, and the adverbs, which require NLPyPort. Those features do not only increase the complexity of the model, but also of Amaia's installation, which will depend on additional software packages. In fact, even though the STS model only requires the dependency parsing and PoS tagging, in order to compute such features, those external libraries end up making additional analysis that takes time, with no direct benefits for the model.

We thus decided to test the impact of removing the aforementioned features from the 27-feature model. However, before this, we analyzed the impact of reducing the size of the largest word embeddings, namely word2vec-CBOW, GloVe, and fastText.cc. This had in mind that, in the embeddings matrix, words of the vocabulary are ordered according to their frequency, i.e., most frequent words are in the initial lines and the final lines include rare words, frequently typos. Thus, what we did was to remove everything except the first 300,000 lines of each of the three aforementioned models and repeat the experiments of Section 4.2 with the smaller versions.

The conclusion was that such a reduction did not impact the performance in AIA-BDE. The highest drop of performance was 0.2 percentual points in fastText.cc, while word2vec-CBOW and GloVe had exactly the same performance and an increase of 0.1 points in the top-5. Therefore, we decided to start using the reduced embeddings, thus decreasing the memory required for the STS model.

After that, we moved on to what can be seen as an ablation study. Table 8 shows the performance of each model when tested on both ASSIN and ASSIN-2 test collections. More precisely, it has the Pearson correlation (ρ) and the Mean Square Error (MSE) between the automatically-assigned similarity scores and those in the collection, which is based on human opinions. In the first line, REDUCED-27 corresponds to the model that uses the 27 features, with subsequent lines corresponding to models where features were manually removed, namely: ADV for the adverbs, DP for dependency parsing, CBOW for word2vec-CBOW, FT for fastText.cc, NB for Numberbatch, and PTLKB for the PT-LKB embeddings. Removing a feature based on embeddings, in fact, entails the removal of both features computed from them, namely, the average embeddings vector and the one weighted with TF-IDF.

Table 8. Performance of the STS model on ASSIN and ASSIN-2 collections, when additional features are removed.

Configuration	**ASSIN 1-PTPT**		**ASSIN 1-PTBR**		**ASSIN 2**	
	ρ	**MSE**	ρ	**MSE**	ρ	**MSE**
REDUCED-27	0.71	0.65	0.71	0.38	**0.75**	**0.54**
R/ ADV, DP	**0.72**	0.65	0.71	**0.37**	0.73	0.58
R/ ADV	**0.72**	**0.64**	**0.72**	**0.37**	0.73	0.58
R/ ADV, FT, PTLKB, GloVe, NB	0.71	0.67	0.71	0.39	0.69	0.64
R/ ADV, CBOW, PTLKB, GloVe, NB	**0.72**	0.66	0.71	0.38	0.71	0.60
R/ ADV, CBOW, FT, GloVe, NB	**0.72**	0.67	0.70	0.39	0.70	0.62
R/ ADV, CBOW, FT, PTLKB, NB	0.71	0.68	0.71	0.39	0.71	0.61
R/ ADV, CBOW, FT, PTLKB, GloVe	0.71	0.66	0.69	0.40	0.69	0.65
R/ ADV, CBOW, GloVe, NB	**0.72**	0.66	0.71	0.38	0.73	0.57
R/ ADV, CBOW, GloVe	**0.72**	0.65	0.71	0.38	0.73	0.57
R/ ADV, DP, CBOW, PTLKB, GloVe, NB	**0.72**	0.66	0.71	0.38	0.71	0.60
R/ ADV, DP, CBOW, PTLKB, NB	**0.72**	0.66	0.71	0.38	0.73	0.58
R/ ADV, DP, PTLKB, GloVe, NB	**0.72**	0.66	**0.72**	**0.37**	0.72	0.59
R/ ADV, DP, PTLKB, NB	**0.72**	0.65	**0.72**	**0.37**	0.73	0.59

The REDUCED-27 model achieved the best performance in ASSIN-2 (ρ = 0.75), but not in ASSIN, where several other models achieved the best Pearson correlation (ρ = 0.72). Performance differences are not substantial. However, we can say that Numberbatch or PT-LKB embeddings do not contribute enough to good performance, and when each of them is the only model of embeddings (lines 6 and 8), performance is generally low. Unlike the others, which have been learned from large quantities of text, these were learned from semantic networks, and thus have lower vocabulary coverage. At the same time, fastText seems to be essential for a good performance in ASSIN-2. Out of the tested models, we selected three for comparison in the AIA-BDE corpus, namely those that we see as a having a good balance between performance and number of used features. None of the selected models uses the dependencies nor the adverbs feature and all use fastText.

The test in AIA-BDE allows for an analysis of the models behavior in a scenario closer to what we expect from Amaia, especially when considering the manually-created variations (VIN, VUC, VMT). As we did in other tables concerning tests in AIA-BDE, Table 9 has the performance of REDUCED-27 and of the selected models when used for matching the original questions and the set of all variations. It shows that the performance of REDUCED-27 and the selected models are not harmed by the reduction of features. Even if by a low margin, the best performing model overall uses GloVe instead of word2vec-CBOW. Its accuracy is 81.3% and 94.1%, respectively, for the first result and for the top-5, which, though not substantial, is still more than one point higher than the unsupervised approaches, based on IR (Table 3) and word vectors (Table 5)

Table 10 has the performance for each variation. The best model overall, in the second row, is also the best for all variations, except for VMT, where the best performance is by REDUCED-27, with all the others tied. When looking at the performance in the top-5, the best model is also not always the same, but differences are low.

Table 9. Proportion of original questions and variations correctly matched with original FAQs (Top1), in Top-3 and in Top-5 most similar, using the most promising reduced STS models.

Approach	**Original (855)**			**Variations (4973)**		
	Top1	**Top3**	**Top5**	**Top1**	**Top3**	**Top5**
REDUCED-27	99.4	**100.0**	**100.0**	80.0	91.4	93.9
R/ ADV, DP, CBOW, PTLKB, NB	98.8	99.6	99.9	**81.3**	**92.0**	**94.1**
R/ ADV, DP, PTLKB, GloVe, NB	**99.5**	99.9	99.9	78.2	90.2	93.2
R/ ADV, DP, PTLKB, NB	98.8	99.9	99.9	79.3	90.6	93.3

Table 10. Proportion of variations of different types correctly matched with original FAQs (Top1), in Top-3 and in Top-5 most similar, using different reduced models.

Approach	**Variation**														
	VG1 (855)			**VG2 (855)**			**VIN (2279)**			**VUC (816)**			**VMT (168)**		
	Top1	**Top3**	**Top5**	**Top1**	**Top3**	**Top5**	**Top1**	**Top3**	**Top5**	**Top1**	**Top3**	**Top5**	**Top1**	**Top3**	**Top5**
REDUCED-27	87.8	95.4	**96.8**	85.9	94.3	**96.4**	81.3	93.5	95.7	64.5	79.3	83.8	**69.1**	**87.5**	91.1
R/ ADV, DP, CBOW, PTLKB, NB	**88.8**	**95.6**	96.5	**86.9**	**94.4**	95.9	**83.5**	**93.9**	95.8	**65.3**	**81.6**	**85.4**	64.3	85.1	**91.7**
R/ ADV, DP, PTLKB, GloVe, NB	84.6	93.6	95.6	82.0	91.9	94.7	82.3	93.3	**95.9**	58.8	77.5	82.7	64.3	83.3	87.5
R/ ADV, DP, PTLKB, NB	85.4	93.5	95.4	82.8	92.3	94.5	82.4	93.4	95.5	63.9	78.4	83.0	64.3	83.9	91.1

5.3. Combining IR with STS

Even with a reduced model of 19 features, relying on a STS model implies that STS is computed between each user interaction and all the questions in the agent's KB. For a large KB, this might result in higher response times.

In the IR alternative, however, this problem is minimized, due to the index. Therefore, in a final experimentation, we aimed at combining the IR approach with the STS model. More precisely, we create an index with the best Whoosh configuration (Stemming + CharsetFilter, see Table 4) and then, for each user interaction, we use Whoosh for retrieving the 30 most relevant questions in the KB, and only apply the STS model to the questions in this subset. While this definitely makes the system more scalable, we had to test whether it could harm performance.

Table 11 has the overall performance of REDUCED-27 and of the best model in the previous section (Table 9). Surprisingly, when considering only the first result, performance is not at all harmed. Together with the results of Section 4.1, this supports that traditional IR is already a good baseline for matching user interactions with questions. Though not always in the first position, it often includes the best candidate in the top retrieved candidates. Moreover, STS is better for discriminating the single best candidate out of the top retrieved. On the other hand, considering the presence in the top-5, performance has a small drop of 0.4 points for the best model and 0.3 for REDUCED-27. However, given that scalability can be significantly increased with the initial selection by the IR approach, with a still neglectable loss of performance, we opted for this combination as the question-matching approach of Amaia.

Table 11. Proportion of original questions and variations correctly matched with original FAQs (Top1), in Top-3 and in Top-5 most similar, using the most promising STS models only on the Top-30 most relevant questions, according to Whoosh.

Approach	**Original (855)**			**Variations (4973)**		
	Top1	**Top3**	**Top5**	**Top1**	**Top3**	**Top5**
REDUCED-27	96.5	98.5	98.6	80.7	91.1	93.6
R/ ADV, DP, CBOW, PTLKB, NB	98.5	99.9	99.9	**81.4**	**91.7**	**93.7**

6. Identifying Out-of-Domain Interactions

A common limitation of IR-based conversational agents is in handling Out-Of-Domain (OOD) interactions. Though not always required, to give the agent a more human-like behavior, it would be interesting to have responses for virtually any question. Therefore, to complement Amaia's capability of answering entrepreneurship questions (domain), we have compiled the Chitchat corpus (see Section 3.2), to be used instead of AIA-BDE, when OOD interactions are identified.

In order to identify OOD interactions and decide whether to search for matching questions in the AIA-BDE or in the Chitchat corpus, a text classifier was trained with all 855 original questions of AIA-BDE (domain) and the same amount of randomly selected interactions from the original Subtle corpus (OOD), for balancing reasons. The Chitchat corpus was not used directly for being too small to provide both training and test data. The performance of the classifier was computed for the classification of OOD interactions in four test sets, each one with all the available variations of each type, namely VG1, VG2, VIN, and VUC. A new random selection of 855 questions from the Subtle corpus was added to each test set. This selection was the same for the four datasets. For an overall performance, a fifth dataset contained all of the 4805 question variations (except VMT, which were not available at the moment), and the same number of randomly selected OOD interactions, again obtained from Subtle.

Three classification algorithms available in the scikit-learn library were tested for this, namely a Linear SVM, a Random Forest classifier (RF) and a Naïve Bayes (NB) classifier, all used with default parameters. For all, questions were represented by their TF-IDF-weighted vectors. Table 12 shows the performance of each classifier, measured with the precision, recall and F1-score of correctly identifying OOD interactions.

Table 12. Performance of different algorithms, when classifying out-of-domain interactions against question variations of different types.

Method		**VG1**			**VG2**			**VUC**			**VIN**			**All**	
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
SVM	96%	93%	95%	96%	93%	92%	95%	93%	94%	92%	93%	93%	96%	86%	91%
NB	95%	88%	92%	95%	88%	92%	96%	88%	92%	91%	88%	90%	96%	80%	87%
RF	97%	96%	96%	97%	96%	96%	95%	96%	96%	95%	78%	86%	95%	79%	86%

Results show that the performance of the classifiers in this task is positive. This happens mostly due to the significant differences between the questions in AIA-BDE and in Subtle (see, e.g., Tables 1 and 2). Nevertheless, performance is slightly lower for the VIN and VUC variations, created manually, than for VG1 and VG2, which have more similarities with the original questions. In any case, the SVM classifier performed better than the other two, with a precision of 96% and a recall of 86% overall. This still means that 14 out of 100 OOD interactions will be incorrectly classified, i.e., will be considered domain questions, and thus their response will be retrieved from AIA-BDE. This should not harm the system too much—at least not as much as the 4 out of each 100 interactions that will be incorrectly classified as OOD. Given that this is a binary classification problem, this means that such interactions were domain questions and that, unless they are rephrased, users will never get their answer. This is why we decided to use the SVM classifier for classifying OOD interactions but, at the same time, included a flag that enables the programmer to turn it off easily.

7. Amaia: A Portuguese Conversational System

Amaia is a Portuguese conversational agent that results from the combination of the previous components. Besides the two corpora (AIA-BDE and Chitchat) used, each indexed in a different Whoosh index, it includes a reduced version of the SVR model for STS, and a SVM classifier of OOD interactions.

In order to get a suitable response R, any interaction I with Amaia goes through the workflow in Algorithm 1. Two parameters are configurable, namely the maximum number of returned questions and answers (n) and a threshold for including a question that is similar to top (θ). We empirically set these parameters to $n = 3$ and $\theta = 0.1$, but, depending on the desired behavior, they can be changed when launching Amaia. The same happens for other options. For instance, handling OOD interactions may be turned off, which makes Amaia always search for the most similar domain question. Whoosh may also be turned off, which implies that the STS model is used for computing STS against all questions in the KB, and not just a subset. This may also be the option for those cases when Whoosh does not retrieve any question for an interaction. However, currently, in this case, Amaia just gives the default response: "Desculpe, não percebi, pode colocar a sua questão de outra forma?' (I'm sorry, I didn't understand, could you rephrase your question?). While it is unlikely to happen with the current configuration (In a Whoosh index of AIA-BDE with the configuration selected in Section 4.1, this happens for four out of the 4973 variations), this behavior works as a fallback mechanism.

The algorithm is complemented with the diagram in Figure 1, which shows the different paths taken by interactions, depending on their classification as OOD or domain, then resulting in different responses. Depending on the classification, a different retriever is used. Moreover, before returning a response, domain interactions are re-ranked according to the STS model.

Algorithm 1: Amaia's workflow.

```
Given an interaction $I$, use the classifier to label it as OOD or domain;
if $I$ is labelled as OOD then
    Search for $I$ in a Whoosh index of the Chitchat corpus;
    Use the response of the first retrieved interaction as $R$;
else
    Search for $I$ in a Whoosh index of AIA-BDE;
    Get the top-30 questions retrieved;
    Compute the STS with each of those questions;
    Build $R$ with the following template: "Se a sua pergunta foi:  <P> R: <R>" ("If your question was <P> R:
      <R>"), with <P> replaced by the question and <R> replaced by its answer;
    For each additional retrieved questions in the top-$n$ for which the difference of the STS is only $\theta$ lower than the
      best, concatenate the following text to $R$: "Também poderá estar interessado em:  <P> R: <R>" ("You may
      also be interested in <P> R: <R>")
end
Give $R$ as the answer.
```

As it is, Amaia can be easily integrated in Slack (https://slack.com/) or any other communication platform with an API that allows for the integration of bots. Figure 2 is a brief real conversation with Amaia that illustrates its capabilities. We highlight how it switches between domain and OOD interactions.

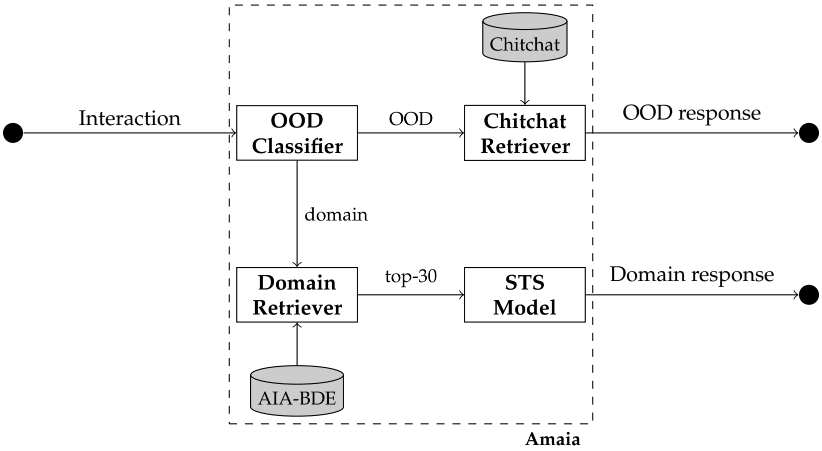
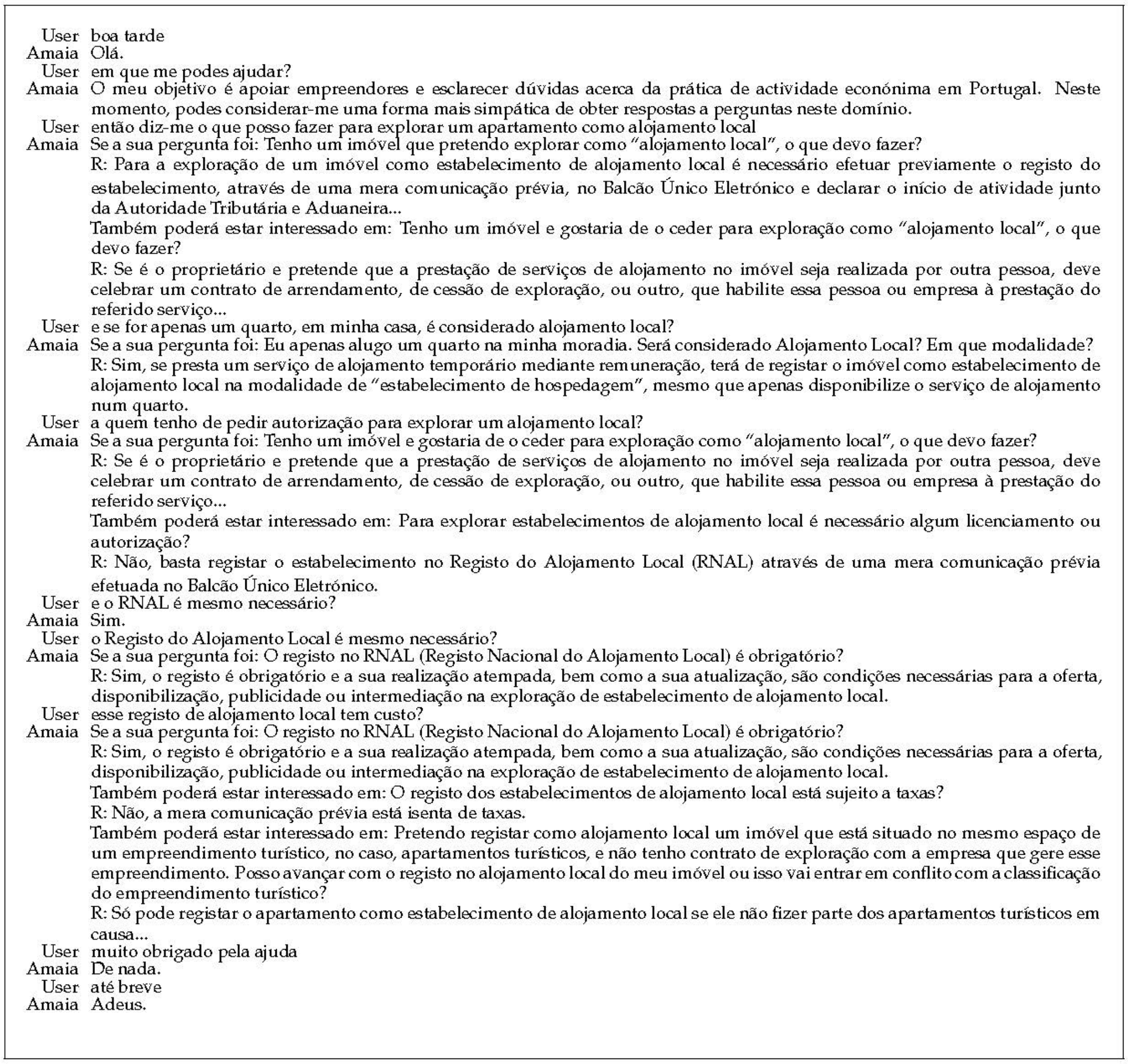


Figure 1. Amaia's workflow.



User	boa tarde
Amaia	Olá.
User	em que me podes ajudar?
Amaia	O meu objetivo é apoiar empreendores e esclarecer dúvidas acerca da prática de actividade econónima em Portugal. Neste momento, podes considerar-me uma forma mais simpática de obter respostas a perguntas neste domínio.
User	então diz-me o que posso fazer para explorar um apartamento como alojamento local
Amaia	Se a sua pergunta foi: Tenho um imóvel que pretendo explorar como "alojamento local", o que devo fazer? R: Para a exploração de um imóvel como estabelecimento de alojamento local é necessário efetuar previamente o registo do estabelecimento, através de uma mera comunicação prévia, no Balcão Único Eletrónico e declarar o início de atividade junto da Autoridade Tributária e Aduaneira... Também poderá estar interessado em: Tenho um imóvel e gostaria de o ceder para exploração como "alojamento local", o que devo fazer? R: Se é o proprietário e pretende que a prestação de serviços de alojamento no imóvel seja realizada por outra pessoa, deve celebrar um contrato de arrendamento, de cessão de exploração, ou outro, que habilite essa pessoa ou empresa à prestação do referido serviço...
User	e se for apenas um quarto, em minha casa, é considerado alojamento local?
Amaia	Se a sua pergunta foi: Eu apenas alugo um quarto na minha moradia. Será considerado Alojamento Local? Em que modalidade? R: Sim, se presta um serviço de alojamento temporário mediante remuneração, terá de registar o imóvel como estabelecimento de alojamento local na modalidade de "estabelecimento de hospedagem", mesmo que apenas disponibilize o serviço de alojamento num quarto.
User	a quem tenho de pedir autorização para explorar um alojamento local?
Amaia	Se a sua pergunta foi: Tenho um imóvel e gostaria de o ceder para exploração como "alojamento local", o que devo fazer? R: Se é o proprietário e pretende que a prestação de serviços de alojamento no imóvel seja realizada por outra pessoa, deve celebrar um contrato de arrendamento, de cessão de exploração, ou outro, que habilite essa pessoa ou empresa à prestação do referido serviço... Também poderá estar interessado em: Para explorar estabelecimentos de alojamento local é necessário algum licenciamento ou autorização? R: Não, basta registar o estabelecimento no Registo do Alojamento Local (RNAL) através de uma mera comunicação prévia efetuada no Balcão Único Eletrónico.
User	e o RNAL é mesmo necessário?
Amaia	Sim.
User	o Registo do Alojamento Local é mesmo necessário?
Amaia	Se a sua pergunta foi: O registo no RNAL (Registo Nacional do Alojamento Local) é obrigatório? R: Sim, o registo é obrigatório e a sua realização atempada, bem como a sua atualização, são condições necessárias para a oferta, disponibilização, publicidade ou intermediação na exploração de estabelecimento de alojamento local.
User	esse registo de alojamento local tem custo?
Amaia	Se a sua pergunta foi: O registo no RNAL (Registo Nacional do Alojamento Local) é obrigatório? R: Sim, o registo é obrigatório e a sua realização atempada, bem como a sua atualização, são condições necessárias para a oferta, disponibilização, publicidade ou intermediação na exploração de estabelecimento de alojamento local. Também poderá estar interessado em: O registo dos estabelecimentos de alojamento local está sujeito a taxas? R: Não, a mera comunicação prévia está isenta de taxas. Também poderá estar interessado em: Pretendo registar como alojamento local um imóvel que está situado no mesmo espaço de um empreendimento turístico, no caso, apartamentos turísticos, e não tenho contrato de exploração com a empresa que gere esse empreendimento. Posso avançar com o registo no alojamento local do meu imóvel ou isso vai entrar em conflito com a classificação do empreendimento turístico? R: Só pode registar o apartamento como estabelecimento de alojamento local se ele não fizer parte dos apartamentos turísticos em causa...
User	muito obrigado pela ajuda
Amaia	De nada.
User	até breve
Amaia	Adeus.

Figure 2. A conversation with Amaia.

The user starts by greeting Amaia ('Good evening') and Amaia says hello, meaning that the interaction was correctly labeled as OOD. In the second interaction, the user asks what Amaia can do to help them, which is again answered with a question in the Chitchat corpus, this time a personal question where Amaia describes its goal. After this, the user asks several domain questions, for which Amaia provides good answers. For the third question (*a quem tenho de pedir autorização...*), the best answer is not the first given, but the second, which supports our option for returning also questions with a close STS. For the fourth question (*e o RNAL é mesmo necessário?*), Amaia's answer is simply "Sim" (Yes). This happens to be a good answer, but only by chance. In fact, the interaction was labeled as OOD. When, in the fifth question, the acronym RNAL is replaced by its full version, the correct answer is given, in the first position. In the final interactions, the user thanks Amaia and says goodbye, with Amaia giving suitable responses (roughly, 'You're welcome' and 'Goodbye').

8. Conclusions

We have described the steps towards the development of Amaia, a conversational agent for helping Portuguese entrepreneurs. After presenting AIA-BDE, the corpus used both as Amaia's KB and as our benchmark, we make an extensive comparison of approaches for matching user requests with existing questions. Those included IR-based approaches, unsupervised STS approaches, and supervised STS models. In the end, we combined the STS model with reduced features, which had achieved the best performance, but only apply it to a subset of the available questions, pre-selected with the best IR-approach. Furthermore, we presented how Amaia uses a text classifier for labeling interactions as domain or OOD, and thus either look for matching questions in AIA-BDE or in a chitchat corpus. Having responses for OOD interactions gives Amaia a more human-like behavior, even if the same interaction has always the same response.

For more variation in the answers, in the future, we may improve how OOD interactions are handled. While learning a generative model could have a negative impact on coherence, we may always define different possible answers for the same question. Moreover, we aim to study how an agent like Amaia may deal with context, and thus avoid giving the same answer several times, while also increasing its performance. This should involve some kind of history, or memory that is updated with each interaction.

The current version of Amaia can be easily integrated in communication platforms, like Slack. In the future, its KB will be increased with more FAQs, which, given our previous options, should not pose challenges on scalability. New FAQs will come from new lists and, ideally, some will be generated automatically, either from structured documents, or from raw text. However, the latter poses a difficult challenge due to the complex language used in most documents we have so far looked at, so additional work is required.

We can say that interesting results were achieved, but there is still much room for improving accuracy. Several improvements may come from alternative ways of combining all the features and/or approaches tested here. For instance, we have not tested promising approaches for STS, namely those based on fine-tuning Transformer neural networks like BERT, which recently achieved high performances for Portuguese [39,40]. Thus far, we just used pre-trained BERT models directly. We also aim to test different combinations of approaches in a voting system, to see whether it is capable of outperforming supervised STS models or not. Finally, some of the approaches could possibly benefit from considering the answers, when matching questions. However, from our preliminary experiments, some of the answers in AIA-BDE are too large and thus an additional source of noise that harms performance.

Author Contributions: Conceptualization, H.G.O. and J.S.; investigation and software, J.S., J.F., and L.D.; methodology, J.S. and H.G.O.; data curation, H.G.O., J.F., J.S., and A.A.; writing–original draft preparation, J.S., H.G.O., and J.F.; writing–review and editing, H.G.O. and A.A.; supervision, H.G.O. and A.A.; funding acquisition and project administration, H.G.O. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by FCT's INCoDe 2030 initiative, in the scope of the demonstration project AIA, "Apoio Inteligente a Empreendedores (*chatbots*)".

Acknowledgments: We would like to thank AMA, especially Jorge Cabrita de Sousa, for following this project and providing us with the FAQs used for AIA-BDE, in addition to Luísa Coheur and her students, for the manual creation of the VIN FAQ variations. We would also like to thank the reviewers of the conference version of this paper.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008.
2. Agirre, E.; Diab, M.; Cer, D.; Gonzalez-Agirre, A. Semeval-2012 task 6: A pilot on semantic textual similarity. In Proceedings of the *SEM 2012: The First, Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation SemEval 2012, Montréal, QB, Canada, 7 June 2012; pp. 385–393.

3. Cer, D.; Diab, M.; Agirre, E.; Lopez-Gazpio, I.; Specia, L. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics, Vancouver, BC, Canada, 3–4 August 2017; pp. 1–14.
4. Fonseca, E.; Santos, L.; Criscuolo, M.; Aluísio, S. Visão Geral da Avaliação de Similaridade Semântica e Inferência Textual. *Linguamática* **2016**, *8*, 3–13.
5. Gonçalo Oliveira, H.; Real, L.; Fonseca, E. (Eds.) Organizing the ASSIN 2 Shared Task. In Proceedings of the ASSIN 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in Portuguese, Salvador, BA, Brazil, 15 October 2019; Volume 2583.
6. Gonçalo Oliveira, H.; Ferreira, J.; Santos, J.; Fialho, P.; Rodrigues, R.; Coheur, L.; Alves, A. AIA-BDE: A Corpus of FAQs in Portuguese and their Variations. In Proceedings of 12th International Conference on Language Resources and Evaluation, Marseille, France, 11–16 May 2020; pp. 5442–5449.
7. Ameixa, D.; Coheur, L.; Redol, R.A. *From Subtitles to Human Interactions: Introducing the Subtle Corpus;* Technical Report; INESC-ID: Lisbon, Portugal, 2013.
8. Santos, J.; Alves, A.; Gonçalo Oliveira, H. Leveraging on Semantic Textual Similarity for developing a Portuguese Dialogue System. In Processing of the Portuguese Language-13th International Conference, PROPOR 2020, Évora, Portugal, 2–4 March 2020; Volume 12037, pp. 131–142.
9. Vinyals, O.; Le, Q.V. A Neural Conversational Model. In Proceedings of the Deep Learning Workshop at ICML, Lille, France, 6–11 July 2015.
10. Voorhees, E.M. The TREC Question Answering Track. *Nat. Lang. Eng.* **2001**, *7*, 361–378, [CrossRef]
11. Rinaldi, F.; Dowdall, J.; Hess, M.; Mollá, D.; Schwitter, R.; Kaljurand, K. Knowledge-Based Question Answering. In Proceedings of the 7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES 2003), Oxford, UK, 3–5 September 2003; pp. 785–792.
12. Kolomiyets, O.; Moens, M.F. A Survey on Question Answering Technology from an Information Retrieval Perspective. *Inf. Sci.* **2011**, *181*, 5412–5434. [CrossRef]
13. Ji, Z.; Lu, Z.; Li, H. An Information Retrieval Approach to Short Text Conversation. *arXiv* **2014**, arXiv:abs/1408.6988.
14. Cui, L.; Huang, S.; Wei, F.; Tan, C.; Duan, C.; Zhou, M. SuperAgent: A Customer Service Chatbot for E-commerce Websites. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 97–102.
15. Agirre, E.; Banea, C.; Cer, D.; Diab, M.; Gonzalez-Agirre, A.; Mihalcea, R.; Rigau, G.; Wiebe, J. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), San Diego, CA, USA, 16–17 June 2016; pp. 497–511; Association for Computational Linguistics.
16. Nakov, P.; Màrquez, L.; Moschitti, A.; Magdy, W.; Mubarak, H.; Freihat, A.A.; Glass, J.; Randeree, B. SemEval-2016 Task 3: Community Question Answering. In Proceedings of the 10th International Workshop on Semantic Evaluation, (SemEval-2016), San Diego, CA, USA, 16–17 June 2016.
17. Nakov, P.; Hoogeveen, D.; Màrquez, L.; Moschitti, A.; Mubarak, H.; Baldwin, T.; Verspoor, K. SemEval-2017 Task 3: Community Question Answering. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 27–48.
18. Kothari, G.; Negi, S.; Faruquie, T.A.; Chakaravarthy, V.T.; Subramaniam, L.V. SMS Based Interface for FAQ Retrieval. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2, Singapore, 2–7 August 2009; pp. 852–860.
19. Karan, M.; Žmak, L.; Šnajder, J. Frequently Asked Questions Retrieval for Croatian Based on Semantic Textual Similarity. In Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing, Sofia, Bulgaria, 8–9 August 2013; pp. 24–33.
20. Caputo, A.; Degemmis, M.; Lops, P.; Lovecchio, F.; Manzari, V. Overview of the EVALITA 2016 Question Answering for Frequently Asked Questions (QA4FAQ) Task. In Proceedings of the 3rd Italian Conference on Computational Linguistics (CLiC-it 2016) & 5th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Final Workshop (EVALITA 2016), Naples, Italy, 12 December 2016; Volume 1749.

21. Pipitone, A.; Tirone, G.; Pirrone, R. ChiLab4It system in the QA4FAQ competition. In Proceedings of the 5th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Naples, Italy, 20 December 2016; Volume 1749.
22. Fonseca, E.R.; Magnolini, S.; Feltracco, A.; Qwaider, M.R.H.; Magnini, B. Tweaking Word Embeddings for FAQ Ranking. In Proceedings of the 5th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Naples, Italy, 20 December 2016; Volume 1749.
23. Magarreiro, D.; Coheur, L.; Melour, F.S. Using subtitles to deal with Out-of-Domain interactions. In Proceedings of the 18th Workshop on the Semantics and Pragmatics of Dialogue (SemDial), Edinburgh, UK, 16–18 June 2014; pp. 98–106.
24. Li, J.; Galley, M.; Brockett, C.; Spithourakis, G.; Gao, J.; Dolan, B. A Persona-Based Neural Conversation Model. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 28 April–1 May, 2016; pp. 994–1003.
25. Melo, G.; Coheur, L. Towards a Conversational Agent with "Character". In Prcoceedings of the Portuguese Language-14th International Conference, PROPOR 2020, Evora, Portugal, 2–4 March 2020; Volume 12037, pp. 420–424, [CrossRef]
26. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
27. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the 1st International Conference on Learning Representations, Scottsdale, AZ, USA, 2–4 May 2013.
28. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543, [CrossRef]
29. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
30. Hartmann, N.S.; Fonseca, E.R.; Shulby, C.D.; Treviso, M.V.; Rodrigues, J.S.; Aluísio, S.M. Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks. In Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology (STIL 2017), Uberlândia, Brazil, 26 May 2017.
31. Řehůřek, R.; Sojka, P. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, 22 May 2010; pp. 45–50.
32. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, 2–7 June 2019; pp. 4171–4186,
33. Souza, F.; Nogueira, R.; Lotufo, R. Portuguese Named Entity Recognition using BERT-CRF. *arXiv* **2019**, arXiv:1909.10649
34. Bird, S.; Klein, E.; Loper, E. *Natural Language Processing with Python*; O'Reilly Media: Sebastopol, CA, USA, 2009.
35. Ferreira, J.; Gonçalo Oliveira, H.; Rodrigues, R. Improving NLTK for Processing Portuguese. In *Symposium on Languages, Applications and Technologies (SLATE 2019), Coimbra, Portugal*; OASIcs, Schloss Dagstuhl: Wadern, Germany, 2019; Volume 74, pp. 18:1–18:9.
36. Speer, R.; Chin, J.; Havasi, C. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In Proceedings of the 3st AAAI Conference on Artificial Intelligence; AAAI Press: San Francisco, CA, USA, 4–9 February 2017; pp. 4444–4451.
37. Gonçalo Oliveira, H. Learning Word Embeddings from Portuguese Lexical-Semantic Knowledge Bases. In Proceedings of the Computational Processing of the Portuguese Language-13th International Conference, PROPOR 2018, Canela, Brazil, 24–26 September 2018; Volume 11122, pp. 265–271.
38. Santos, J.; Alves, A.; Gonçalo Oliveira, H. ASAPPpy: A Python Framework for Portuguese STS. In Proceedings of the ASSIN 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in Portuguese co-located with XII Symposium in Information and Human Language Technology (STIL 2019), Salvador, Brazil, 15 October 2019; Volume 2583, pp. 14–26.

39. Rodrigues, R.; Couto, P.; Rodrigues, I. IPR: The Semantic Textual Similarity and Recognizing Textual Entailment Systems. In Proceedings of the ASSIN 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in Portuguese co-located with XII Symposium in Information and Human Language Technology (STIL 2019), Salvador, BA, Brazil, 15 October 2019; Volume 2583, pp. 39–48.
40. Fonseca, E.; Alvarenga, J.P.R. Multilingual Transformer Ensembles for Portuguese Natural Language Tasks. In Proceedings of the ASSIN 2 Shared Task: Evaluating Semantic Textual Similarity and Textual Entailment in Portuguese Co-Located with XII Symposium in Information and Human Language Technology (STIL 2019), Salvador, Brazil, 15 October 2019; Volume 2583, pp. 68–77.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).