February 2023

Assisting the Assistant: a Cobot for Voice Customer Support

Mario CORRADO [a] Vincenzo GILIBERTI [a] Manuel GOZZI [b] Vincenzo LANZOLLA [a]
Guido VETERE [b,c] Domenico ZURLO [a]
[a] *IN & OUT S.p.A. Single Shareholder Teleperformance S.E., Taranto, Italy*
[b] *Isagog S.r.l., Roma, Italy*
[c] *Università Guglielmo Marconi, Roma, Italy*
ORCiD ID: Guido Vetere https://orcid.org/0000-0002-6703-7276

Abstract. Despite recent advances in automation, customer support still require a substantial amount of human intervention through voice channels. With the aim of improving the work of human assistants, we developed [1] a collaborative bot (cobot) to help them in the process of handling customer voice interactions. The cobot is a reasoning agent that starts from loading background customer data into a dynamic knowledge graph. Then it captures the audio stream of the conversation, converts it to text in real time, analyzes the blocks of conversation with neural technologies and "thinks" about the results. Assistants can also supply data to the cobot, based on the information they gather from the ongoing conversation. The reasoning agent provides information and action suggestions to the human assistant by applying heuristics on data collected from both automatic and human sources, based on a task and domain-specific conceptual models (ontologies). While designing a prototypical solution for utility services in Italy, we face with many problems, including spontaneous speech understanding, factual and linguistic knowledge representation, and efficient heuristic reasoning. We adopted a standards-based approach and experimented with open source reasoners and publicly available language models. The paper presents preliminary findings and outlines the system design, with focus on the interplay of neural language processing and logic reasoning.

Keywords. virtual assistants, colaborative bots, knowledge graph, ontologies, natural language understanding

1. Introduction and Background

The application of intelligent virtual assistants (IVA) to complex tasks, such as helping customers with solving problems or completing administrative procedures, is perceived as a critical business need. However, although IVAs can perform calculations, conduct seemingly human-like conversations, provide answers to questions formulated in natural language, perform actions on users' devices, many customer cases still require human intervention [15]. A common strategy for backing virtual assistants while retaining the best of automation is *fail-over*: when the virtual assistant gives up trying to handle the transaction, it activates human operators. But this strategy isn't without its pitfalls: users may get frustrated by the waste of time, and any bot failure can be detrimental to the company's reputation. The project `VO.I.C.E. First` is about overcoming these problems by adopting a different approach: instead of engaging customers in potentially useless automated dialogues, our system supports human assistants while interacting with them, with the aim of saving interaction time and increasing the service quality. The benefits we expect are particularly relevant in a country with an elderly population and poor digital skills such as Italy. Pursuing this approach means designing and building a sort of "collaborative bot" (*cobot*), which can be regarded to as a "cognitive co-worker".

Human-robot collaboration has been mainly studied from the point of view of coexistence and cooperation in physical environments [4]. In such contexts, the cognitive abilities of the robot mostly consists in environmental awareness and knowledge of manufacturing workflows [6]. Interacting by vocal commands with such devices is crucial in many application scenarios [1]. Vocal interfaces have been also experimented to dynamically change the cobot behavior [11]. For humans and machines working together on pure cognitive tasks, however, semantic understanding of speech cues cannot be limited to recognizing a set of requests and instructions, but must be able to explore relevant bodies of knowledge. A number of systems have been developed to provide this ability. IBM COBOTS, for instance, orchestrates an ensemble of "Domain Expert" bots which rely on Knowledge Graphs of various system management tasks [27]. However, most of these systems only support text-based interactions [19].

Developing a voice-enabled cognitive cobot faces a number of challenges. Automated text conversion of the customers' speech must take place in real time, which sometimes requires an arbitrary segmentation of their utterances. Spontaneous speech features syntactic incoherence, interjective and phatic expressions, which unfold in many turns of intricate conversations. It's quite hard to frame this kind of utterances into predefined patterns (skills), such as those underlying popular vocal assistants, e.g. Alexa [10]. Still, in order to provide the assistant with helpful suggestions about clients' situation, it is necessary for it to achieve some "understanding" of their intentions and needs.

To cope with these challenges, we integrate knowledge representation and machine learning methods. Broadly speaking, the system reasons at the "best explanation" (abduction) of the users' utterances, with respect to background knowledge and some understanding of the customer situation as it emerges in the conversation. On the one hand, this approach relies on an accurate modeling of general, domain-specific, and task-specific concepts; on the other hand, the users' spontaneous speech is analyzed by neural systems based on large language models (LLM) [17]. This approach differs from those based on machine learning alone, and comes at the price of coping with knowledge representation. However, we have considered that obtaining fine-grained knowledge by applying statistical methods (albeit sophisticated) on corporate data, whether structured or linguistic, would still require non-trivial adaptation (tuning, prompt engineering) efforts, as well as an accurate verification of results. Still, we want to leverage the power of new Natural Language Processing (NLP) tools and resources, including open sourced ones [2]. The focus of our project is therefore the search for an adequate balance between learning and representation methods. We pursue our goal by delimiting the scope of neural NLP and integrating it with logical reasoning. As such, our approach heads towards neuro-symbolic systems [23]

Research shows how structured knowledge (Knowledge Graphs, KG in the sequel) can be used to improve automatic recognition and language processing. When compared

[2] In particular, Huggingface transformers `https://huggingface.co/docs/transformers`

with neural-based recognition of preconfigured intents, these knowledge bases, together with suitable reasoning procedures, can fuel the ability to “understand” users’ expressions in accurate and flexible way[12]. Flexibility here means the possibility of enriching the system’s linguistic competence without having to resort to training processes. To support conceptual modeling, we embraced W3C’s Semantic Web (SW) standards [7]. Beyond the general aims of the SW, these standards (namely: RDF, OWL, SWRL and SPARQL) offer a complete set of knowledge representation, reasoning and querying formalisms, all of which can be provided by Enterprise KG platforms [21]. Based on a set of ontologies deployed on a standard KG platform, we design, implement and execute rules that link background data, customer utterances, and human findings, to obtain a system that provides information and suggests actions. Most of the challenge we face lies in these heuristic rules and how to properly activate them based on low-quality input.

2. System Overview

When the assistant answers a call and the customer is identified, the relevant customer information is loaded into an ephemeral in-memory KG which will be the cobot’s dynamic interpretation of the conversation. Customer data can be obtained from a CRM database through an appropriate mapping to the cobot ontology, which is also loaded in the ephemeral KG at the start-up. The cobot will then add assertions to the KG based on input received from a NLP pipeline parsing user utterances, as well as information provided by the assistant via an internal chat. At each update, the cobot will jointly reason on the customer data as received from the CRM, the customer expressions as interpreted by the NLP pipeline, and the supplementary information it may collect from the assistant.

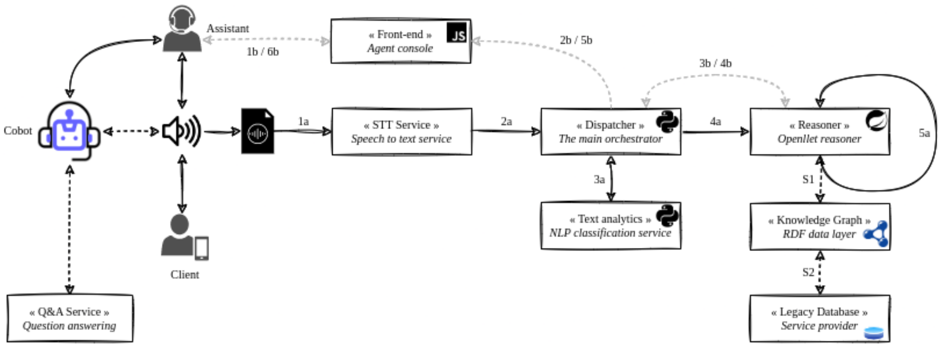


Figure 1. VO.I.C.E. First Overview

Table 1. Legend

Step	Description
1a	Assistant and customer talk. A listening service captures the audio stream and sends it in real time to the STT service
2a	The STT service converts the audio into natural text and propagates the result to the dispatching service
3a	The dispatching service forwards the input to the text classification service, which identifies the keywords that categorize the act of speech in progress.
4a	The dispatching service sends the classification result to the reasoning service
5a	The reasoning service translates the classification results in a suitable update to the KG
1b	During the whole phone call, the assistant uses a console that summarizes the state of the speech
2b	The console ”asks” the dispatching service for the presence of inferences that can be deduced from the content of the graph itself.
3b	The dispatching service forwards a request to the reasoning service with the reference to the KG. The reasoning service receives the request and launches a SPARQL query to compute the inferences, thus starting the reasoning process.
4b	The output of the reasoning process is encoded and sent back to the calling service.
5b	The dispatching service propagates the result up to the console.
6b	Once the reasoning result is received by the console, a message for the Assistant is produced in various forms (pop-up, windows opening)

3. Ontology

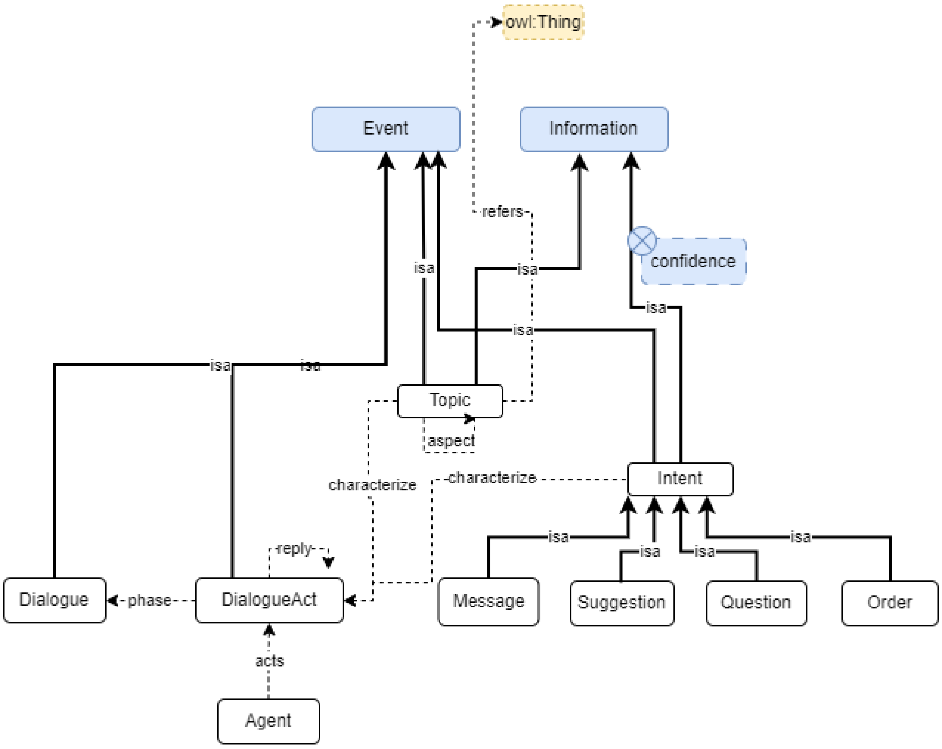


Figure 2. Dialogue modeling in `VO.I.C.E. First` upper level.

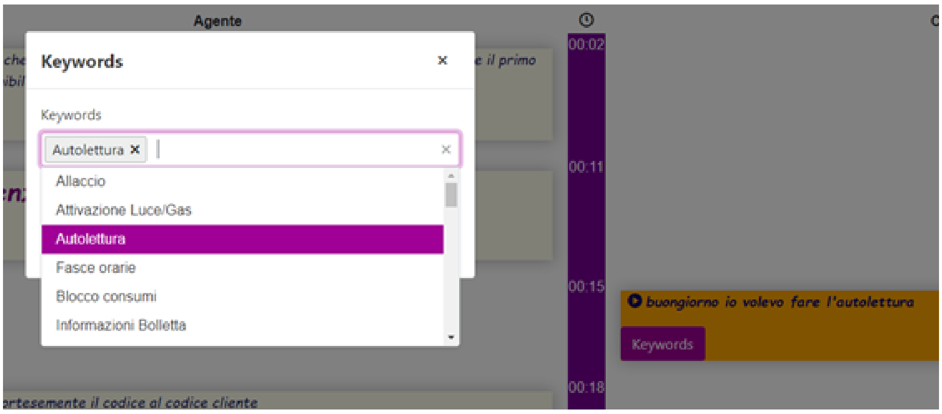
To support inference, `KGs` must be built on conceptual schemes (ontologies) that basically consist in "meaning axioms" specified by some description logic, possibly supplemented by rules. The quality of these conceptual schemes proves to be crucial for the overall functionality of the system. We designed an "upper-level" of few general categories and relations, starting from basic distinctions of well established standards [8] [2]. In the line of previous works, such as [28] and [25], we also modeled dialogue acts and their relations with referred entities. As shown in Figure 2, each interaction (`Dialogue`) collects the interwoven speech acts (`DialogueAct`) of three agents: the customer, the assistant and the cobot. `DialogueActs` are characterized by an `Intent` and composed by some `Topic`. `Intents` reflect the pragmatics of the act (purpose), specifically in the perspective of performativity [3], while `Topics` carry the semantic workload. `Intents` and `Topics` feature a `confidence` property, that makes the uncertainty of the `NLP` pipeline available to heuristic reasoning. `Topics` can refer to any KG entity, and can relate to other topics as `aspects`. A distinguishing feature of our upper level is a non-categorical account of conceptual roots, i.e. top level concepts are not disjoint. In our case, `Topics` and `Intents` are subsumed to both `Information` (*non-physical continuant*) and `Event` (*temporal occurrent*), as they act as "semiotic objects", i.e. physical entities which convey conceptual (abstract) meanings.

4. Linguistic Analysis

Spontaneous speech analysis is a classic challenging task [9]. The conversational speech we are faced with presents specific difficulties, as it is often affected by dispersion, noise and incoherence. To interpret users' utterances (in Italian language), we had to tailor a specific pipeline. The goal is to identify blocks of conversation ("chunks") and analyze them in terms of pragmatics (`Intent`), and semantics (`Topic`). As for intents, we limit the analysis to the basic distinction among "ask" and "tell", i.e., respectively, information request and supply. The most frequent topics in real conversations were analyzed in the light of out ontology; we also introduced new concepts when needed. `Intents` and `Topics` classes are annotated to facilitate the integration of the `NLP` pipeline with the reasoning system.

4.1. The Corpus

Our study was conducted starting from an analysis of the application domain of the energy sector, through 3000 transcripts of real conversations provided by an utility company, using a proprietary web application called *TP Tagger*. This application allowed us to obtain the transcription of audio files of telephone conversation recordings, providing features to anonymize transcriptions, erase sensitive data, and adding metadata (Figure 3), including a coarse classification of the conversation purpose. From this base, we choose 2151 meaningful fragments (i.e. containing interesting content) of user's conversation and annotated them with intents and topics. We thus obtained a training set to feed a number of classification tasks.



il primo

buongiorno io volevo fare l'autolettura

rtesemente il codice al codice cliente

Figure 3. TP Tagger annotation.

4.2. Chunking

We had to find a strategy for segmenting user expressions observed in the corpus. Chunking (also known as *shallow parsing*) is the process of splitting sentences into flat list of syntactically unitary segments. As for Italian language, this technique has been extensively experimented in the past decades [16]. However, we realized that what has been developed for written texts is both unnecessary and difficult to adapt to the linguistic material we were dealing with. Indeed, we observed that conversational turns are mostly paratactic, i.e. they don't show much constituent nesting. In particular, the type of chunks we identified as useful seemed to be wider than those typically obtained from the classic shallow parsing techniques. Hence, we have devised a simple approach to split utterances based on coordinated conjunctions. To this aim, we used Stanford's `STANZA` [22] parser, and just partitioned the dependency graph on the `conj` (conjunction) edge. This very simple heuristic has proved to be effective both for analyzing the corpus and the user's expression in real time.

4.3. Annotation

A team of three experts was involved in the annotation of the chosen conversation chunks. For each segment, one out of two intents (interrogative vs. informative) and one or more out of 37 topics related to utility services (mostly regarding billing issues) were independently marked, with an inter-annotator agreement of 78% on intents and 69% on topics. The gold dataset was then constructed by considering the most commonly agreed upon annotation choices, and arbitrating or trimming the edge cases. From this basis, by balancing the number of positive example of each class, we obtained training (actually, tuning) and test datasets for both intent and topic classifiers.

4.4. Classifiers

To build a classifier with the dataset we had available, we adopted a transformer based approach [18]. We experimented with the following multilingual linguistic models (supporting Italian):

- **mDeBERTa**: This multilingual model can perform natural language inference (NLI) on 100 languages and is therefore also suitable for multilingual zero-shot classification [13].
- **GPT3 Davinci**: The most capable GPT3 (Generative Pretrained Transformer 3rd Generation) model. This linguistic algorithm harnesses the power of machine learning to perform various NLP tasks like translating texts, answering questions and is also capable of writing text using its impressive predictive capabilities. GPT-3 works by trying to predict text based on input provided by users [5]. It can perform any task that the other models can perform but expensive and slower than the other GPT3 models and generally the other tested models. Used specifically for the chunk prediction phase given the high capacity and high cost.
- **GPT3 Curie**: high performance model, cheaper and faster, it is the best performing GPT3 model after the Davinci one but also cheaper and faster than the latter.
- **Bert-base-italian** (dbmdz/bert-base-italian-cased): model of the HuggingFace repository made by the MDZ Digital Library team in Italian whose data consists of a recent Wikipedia dump and various texts from the OPUS corpora collection [24].
- **BERTino** (indigo-ai/BERTino): Italian HuggingFace DistilBERT repository model pre-trained by indigo.ai on a large Italian general domain corpus that combines both good performance and small weight compared to other models [20].

Model	Intent	Topic
mDeBERTa	71.19	N/A
GPT3 Curie	90.68	82.05
GPT3 Davinci	91.53	82.05
Bert-base	88.98	69.23
BERTino	90.68	71.79

Table 2. Classifier performance

Based on these experiments, we decided to base our development prototype on a `BERTino` on premise deployment. Although `GPT3 Davinci` showed better results, we considered, for a real-time system, the problem of relying on a remote service. An example of the output of the chunk-classify pipeline is in the following box.

sì salve buongiorno vorrei chiedere un'informazione mi è appena arrivata una bolletta da cinque cento euro[a]

chunk	classification
si, salve buongiorno vorrei chiedere un'informazione	INTENT: INFORMATION_REQUEST
mi è appena arrivata una bolletta da cinque cento euro	TOPICS: INVOICE, AMOUNT

[a] yes hi good morning I would like to ask for information I have just received a bill for five hundred euros

As they exit from the pipeline, the classified chunks are transformed in `KG` updates, each of which causes the reasoning system to refresh the "epistemic state" of the cobot.

5. Reasoning

Reasoning at the "best explanation" of the conversation content is obtained by `Description Logic` (`DL`) inference (in particular, *instance checking*) and heuristic rules, operating on ontology definitions. We adopted Openllet [3], an open source implementation of the Pellet `DL` reasoner [26], which supports `SWRL` [4] rules for hybrid reasoning. We evaluated the expressiveness of `SWRL` as adequate to represent our heuristic rules [14]. The following box sketches one of these rules. Note that a number of instances including `DIALOG`, `COBOT`, `CUSTOMER`, predefined messages such as `SELF_READING_MSG` (suggest a "self reading" action) and constants (e.g. "ESTIMATED") are part of the application ontology, thus are defined in every `KG` instance. Moreover, the update procedure guarantees that such structural `KG` objects are properly linked to dynamic dialog acts.

[3] `https://github.com/Galigator/openllet`

[4] `https://www.w3.org/Submission/SWRL/`

```
DialogueAct(?act) AND Information(?intent)
AND Invoice(?topic) AND Amount(?topic)
AND characterize(?intent, ?act)
AND characterize(?topic, ?act) AND
last_invoice(CUSTOMER, ?invoice) AND
reading_type(?invoice, "ESTIMATED") →
tell(COBOT, SELF_READING_MSG)[a]
```

[a] The rule encodes the hypothesis that the cause of the customer's amount inquiry is a bill based on estimate rather than actual consumption

After each update and subsequent reasoner's run, a query is issued to get the COBOT generated messages. To start, we experimented with real conversation scripts chosen among the simplest ones, and we measured the reasoner's performances in view of its real-time usage. On a Intel Core i7-12700K 3.6 GHz 12-Core Processor, the reasoner takes less than 3 seconds to perform the update, and less that 1 second for queries. We consider these figures encouraging for further experiments on more complex cases, as we are confident that the main complexity factor, i.e. conversation data, is bound to definite limits.

6. Conclusion and Future Work

We developed a hybrid system in which the AI supports human operators instead of trying to replace them. The system integrates human and artificial understanding of both structured and linguistic data, to improve the support of people who cannot use technologies, or do not want to give up contact with human operators. As such, we believe our research has social import. Although still in a preliminary stage, our experience offers a number of interesting technical suggestions. The right combination of neural and symbolic approaches, based on open source software and standards, allows the implementation of effective business solutions, which neither neural nor symbolic approaches alone could address. More work needs to be done to test our system in real situations. Apart from technical improvements and optimizations, there are substantial aspects that need to be better understood. For instance, if the customer provides crucial information while the cobot is reasoning, the system could give useless or even misleading suggestions to the assistant. The impact of synchronization aspects should be carefully evaluated in the future. But there are also deeper aspects to investigate. One of them is the interplay of reasoning and language understanding. In our system, the two processes are separated and sequenced. In particular, the NLP outcome drives the logic inference. We are aware that in the real life, on the contrary, language understanding and reasoning are deeply intertwined. How to feed back on the functioning of neural linguistic models based on symbolic inference processes is an open problem of today's AI.

References

[1] Alexandre Angleraud, Amir MehmanSefat, Metodi Netzev, and Roel Pieters. Coordinating shared tasks in human robot collaboration by commands. *Frontiers in Robotics and AI*, 8, 2021.

[2] Robert Arp, Barry Smith, and Andrew D. Spear. *Building Ontologies with Basic Formal Ontology*. The MIT Press, 2015.

[3] John Langshaw Austin. *How to do things with words*. William James Lectures. Oxford University Press, 1962.

[4] Paul Baxter, Joachim Greeff, and Tony Belpaeme. Cognitive architecture for human–robot interaction: Towards behavioural alignment. *Biologically Inspired Cognitive Architectures*, 6:30–39, 10 2013.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[6] Alejandro Chacón, Pere Ponsa, and Cecilio Angulo. Cognitive interaction analysis in human–robot collaboration using an assembly task. *Electronics*, 10(11), 2021.

[7] W3C Consortium. Semantic web. `https://www.w3.org/standards/semanticweb/`, 2015. 2023-02-26.

[8] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. *Sweetening Ontologies with DOLCE*, pages 166–181. Springer, Berlin, Heidelberg, 2002.

[9] Awni Hannun. The history of speech recognition to the year 2030. *arXiv preprint arXiv:2108.00084*, 2021.

[10] Amazon Inc. Alexa developer documentation. `https://developer.amazon.com/en-US/docs/alexa/documentation-home.html`, 2023. 2023-02-26.

[11] Tudor B. Ionescu and Sebastian Schlund. Programming cobots by voice: A human-centered, web-based approach. *Procedia CIRP*, 97:123–129, 2021. 8th CIRP Conference of Assembly Technology and Systems.

[12] Ashwini Jaya Kumar, Sören Auer, Christoph Schmidt, and Joachim Köhler. Towards a knowledge graph based speech interface. *CoRR*, abs/1705.09222, 2017.

[13] Moritz Laurer, Wouter van Atteveldt, Andreu Salleras Casas, and Kasper Welbers. Less annotating, more classifying. addressing the data scarcity issue of supervised machine learning with deep Transfer learning and bert. *Preprint*, June 2022. Publisher: Open Science Framework.

[14] Abba Lawan and Abdur Rakib. The semantic web rule language expressiveness extensions-a survey. *CoRR*, abs/1903.11723, 2019.

[15] Leah. What do your customers actually think about chatbots? `https://www.userlike.com/en/blog/consumer-chatbot-perceptions`, 2022. 2023-02-26.

[16] Alessandro Lenci, Simonetta Montemagni, and Vito Pirrelli. Chunk-it: An italian shallow parser for robust syntactic annotation. 01 2001.

[17] Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *CoRR*, abs/2111.01243, 2021.

[18] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: A comprehensive review. *ACM Comput. Surv.*, 54(3), apr 2021.

[19] Quim Motger, Xavier Franch, and Jordi Marco. Software-based dialogue systems: Survey, taxonomy and challenges. *ACM Computing Surveys*, 55, 04 2022.

[20] Matteo Muffo and Enrico Bertino. Bertino: An italian distilbert model. In *CLiC-it*, 2020.

[21] J.Z. Pan, G. Vetere, J.M. Gomez-Perez, and H. Wu, editors. *Exploiting Linked Data and Knowledge Graphs for Large Organisations*. Springer, 2017.

[22] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online, July 2020. Association for Computational Linguistics.

[23] Md. Kamruzzaman Sarker, Lu Zhou, Aaron Eberhart, and Pascal Hitzler. Neuro-symbolic artificial intelligence: Current trends. *CoRR*, abs/2105.05330, 2021.

[24] Stefan Schweter. Italian bert and electra models, November 2020.

[25] J. Silva, D. Melo, and I. Rodrigues. An ontology based task oriented dialogue. *In Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge*

Management (IC3K 2021), 2.

[26] Evren Sirin, Bijan Parsia, Bernardo Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *SSRN Electronic Journal*, 01 2007.

[27] Sethuramalingam Subramaniam, Pooja Aggarwal, Gargi Dasgupta, and Amit Paradkar. Cobots - a cognitive multi-bot conversational framework for technical support. 07 2018.

[28] Michael Wessel, Girish Acharya, James Carpenter, and Min Yin. *OntoVPA: An Ontology-Based Dialogue Management System for Virtual Personal Assistants: 8th International Workshop on Spoken Dialog Systems*, pages 219–233. 01 2019.