Journal of Systemics, Cybernetics and Informatics (2022) 20(1), 249-269
https://doi.org/10.54808/JSCI.20.01.249

Information Retrieval based on Brazilian Portuguese Texts

Victor Hayashi[1], Mateus Carvalho[2], João Carlos Néto[1], Felipe Pinna[1], Rosangela Marquesone[1],Wilson Ruggiero[1] and Maisa Duarte[2]

[1]Polytechnic School of the University of São Paulo (USP), Brazil
[2]Bradesco Bank, Brazil

Abstract[1]

Knowledge-based intelligent systems might be used in the banking sector to automate customer service. One of the ways to represent knowledge that is both understandable by humans and readable by machines is by using ontologies[2]. Whenever a customer queries its bank regarding specific products or services, the existing knowledge modeled in an ontology might be used by a customer service chatbot to answer it in an automated way. The existing manual information retrieval process from banking specialists is laborious and time-consuming. Specialists use natural language, visual representations, and common sense, often overlooking details. It is a great challenge to make a specialist's knowledge explicit, formal, precise, and completely scalable, which is the format required by a customer service chatbot. We propose a semi-automatic approach to retrieving banking information in Brazilian Portuguese texts with minimal specialist support. By combining Natural Language Processing techniques (e.g., syntactic analysis to obtain the logical meaning of sentences based on rules and its structure) and an ontology constructor library[3], it was possible to build a tool that receives texts from the banking domain and constructs an ontology that knowledge-based intelligent systems can use. Specialist support is only needed in intermediate refinement steps, thus optimizing the banking specialist's time. The use cases for investments, opening a banking account, and the comparison of the proposed approach show how we reduced manual labor in the information retrieval process by a factor of 40%. Our approach can identify more information in each sentence compared to a similar method found in the literature. The resulting ontologies can be used in a chatbot that automates customer support for a large Brazilian bank.

Keywords: *Information Retrieval, Natural Language, Brazilian Portuguese, Banking, Information Systems.*

* Contact Author: Victor Hayashi (victor.hayashi@usp.br).
[1] Proof reading performed by M. Cristina V. Borba, an experienced Portuguese-English academic translator.
[2] We consider an ontology as "an explicit knowledge level specification of a conceptualization, which may be affected by the particular domain and task it is intended for" (Van Heijst, 1997).
[3] We used the Python programming language library OwlReady2, found in https://owlready2.readthedocs.io/

1. Introduction

Banking services such as investing, lending, transferring money, and paying bills perform an essential role in modern society. Whereas we had to use physical bank branches to perform financial transactions in earlier times, we can now use smartphones and computers anywhere and anytime for those tasks. This transition from physical to virtual changed the financial industry as we know it, reducing costs and enabling automation at large.

However, many clients still rely on human customer service to understand the various banking products and services. While it may be more natural for the customers to interact with human employees, it is a costly alternative for the banks. A novel approach that some businesses are deploying is automated customer service using chatbots. We can understand chatbots[4] as automated agents that can understand and respond to customers' queries in natural language.

Although chatbots may drive cost reductions related to customer support, these agents rely on particular knowledge from the banking domain. When a bank starts to automate customer service using chatbots, it must acquire the knowledge of its products and services in a process known as Knowledge Engineering (Guarino, 1997).

The knowledge related to banking products and services may be present in documents or tacit in banking employees' knowledge. If the knowledge is tacit in banking employees' knowledge, it is not easy to make it explicit. These specialists use natural language, visual representations, and common sense, usually overlooking details implicit in human communication (Tecuci, 2016). Thus, the specialist knowledge form is different from how the knowledge must be represented to be useful for information systems, which are formal, precise, and complete (Debenham, 2012; Kendal, 2007).

[4] Chatbots are user interfaces that allow interactions by voice and text.

We can use structures known as ontologies to model such specific knowledge because of their popularity and support for an evolutive domain knowledge acquisition (Wouters, 2000). Other benefits of using ontologies are automated validation, logical inferences, and facilitated integration with other systems (Happel, 2006). Some examples of using ontologies for the financial sector: Financial Industry Business Ontology (FIBO) was proposed to define financial concepts without ambiguity (Bennett, 2013), and another work used ontologies to represent knowledge about financial products and their clients and identify where the services provided can be improved (Tang, 2011).

Ontologies may be understood as a kind of mind map that is used to describe the concepts of a domain (graph nodes) and how they are related (graph edges). Considering that in this work we must develop ontologies for conversational agents, the ontology is a structured graph that must provide the agent with patterns that lead to an adequate response. We present one example: if the chatbot must answer the question "is banana a fruit?", then it can rely on an ontology that has two nodes: fruit and banana, and a relation "is-a" which indicate that banana is a kind of fruit.

Some manual approaches to building ontologies are slow and laborious, resembling an art (Jones, 1998). In this process, the specialist must communicate his/her knowledge to the Knowledge Engineer, who will transform it into a machine-understandable format. Afterward, the specialist must analyze whether the knowledge base content follows reality (Shaw, 1990; Dudycz, 2015; Gangemi, 2006).

Although these steps are considered essential in Knowledge Engineering, they are complex to be executed. Failing to achieve the necessary information may lead to an incomplete conversational agent that may not know how to answer particular questions. Moreover, consider how much effort would be necessary to obtain all the knowledge of a large bank's products and services contained in documents and tacit in specialists: it would not be feasible at all.

Therefore, we propose a semi-automatic ontology development approach based on natural language descriptions (i.e., texts) in Brazilian Portuguese to improve the knowledge base for customer service chatbots. As inputs, we ask specialists to express their knowledge in textual form or use manuals and other documents from the banking domain. Our objective is to optimize the specialist's time by automating laborious steps in the ontology development process, thus unburdening the specialist of operational tasks so that this employee can deal with more value-adding tasks, such as decision making.

The paper is organized as follows: Section 2 describes the manual process of creating ontologies. Section 3 contains our proposed semi-automatic scheme. We describe banking use cases and a comparison of the proposed method with the manual process in Section 4. A comparison with related work found in the literature is presented in Section 5, and Section 6 provides the final considerations and directions for future work.

2. The Manual and Laborious Way

The development of an ontology is not deterministic: it may have more than one relevant result, and different specialists in knowledge representation have distinct opinions about the best way to model concepts (Tudorache, 2008). Furthermore, even though an ontology can be used for more than one purpose, the decision of using one ontology or another must be based on how this knowledge is used by other systems. In this sense, ontologies have different levels of generalization, abstraction, and conceptualization, and, therefore, it is common to divide them into four categories: task ontology, domain ontology, representation ontology, and generic ontology (Bimba, 2016).

In the present study, the main focus was the development of ontologies aimed at conversational agents; therefore the inputs are texts describing Banking products and services, and the outputs are ontologies that model specialist knowledge that the chatbots can use to answer customer queries. For example, when a client wants to

know the taxes associated with a specific Brazilian investment, the chatbot can use an investment ontology to answer the client, and this investment ontology is based on documents that describe Brazilian investments taxes.

Ontology development has a logical complexity comparable to the software development process. Therefore, diverse methodologies have been proposed in the literature to ensure its quality.

Agile Methodology for Ontology Development (AMOD) uses agile development principles and guidelines to provide a complete methodology that supports relevant details for the steps needed in the ontology life cycle (Abdelghany, 2019). The AMOD methodology classifies various ontology usage activities in the pre-game, development, and post-game Scrum phases:

- Pre-game: objective and scope definition (e.g., who the ontology users are), selection of tools and techniques, ontology requirements specification (e.g., questions that the ontology must support), information source selection;
- Development: the ontology owner selects tasks for knowledge engineers in the sprint planning, and the relevant terms for the specific domain are captured in the knowledge acquisition. Some intermediate conceptual models are created based on the knowledge obtained, and a final formal model is obtained next. Verification and validation steps ensure that the ontology construction is correct and that the apropriate ontology is being constructed. Ontology owner and knowledge engineer revise and analyze the sprint in a final meeting;
- Post-game: the new ontology developed is integrated into the other ontologies developed previously, and the results are registered, which compromises human-understandable, machine-readable, and additional documentation files made available as web resources.

Methodology 101 is a guide for ontology development, and it deals with complex problems related to class hierarchy definitions and class/instances properties (Noy N. F., 2001a). This methodology is based on fundamental rules that support decision-

making in the ontology construction process. It starts with an ontology draft, refined after each iteration, to develop a complete ontology incrementally. Some premises of methodology 101 are:

- There is no unique way to model concepts of a specific domain;
- The ontology development process is necessarily iterative;
- An ontology represents reality; thus, the concepts portrayed in it must abide by this reality.

Methodology 101 can be summarized in six steps:

1. Domain definition and ontology scope;
2. Reuse of existing ontologies;
3. Ordering of terms respective to their importance within the ontology;
4. Class hierarchy definition;
5. Class properties definition;
6. Instances creation.

This article manual process is a combination of the methodologies 101 and AMOD found in the literature, as we consider that neither process combines agile development methods and the reuse of existing ontologies to create a new one. As illustrated in Fig. 1, our approach consists of the following steps:

1. Specific Domain Scope Definition: based on the pre-game phase from AMOD, and step 1 from methodology 101 (e.g., select the fruits domain);
2. Existing Ontologies Analysis: it is imperative to consider the reuse of existing ontologies, according to step 2 of methodology 101 (e.g., search for existing ontologies regarding fruits);
3. Main Terms Enumeration: similar to AMOD development phase and step 3 of methodology 101 (e.g., select the most important concepts of fruits);

4. Classes, Instances, and Hierarchy Definitions: based on AMOD development phase and steps 4 and 6 of methodology 101 (e.g., model banana as a Subclass of the fruit Class);
5. Terms Properties: based on step 5 of methodology 101 (Class Properties Definition) (e.g., include additional information regarding the banana Subclass, such as its origin);
6. Necessary Data + Relations: an additional step that integrated with the previous two steps represent the iterative nature of ontology development described by one of methodology 101 guidelines (e.g., if an existing ontology was used, combine it and the new concepts described);
7. Rules and Restrictions Definition: additional step associated with refinement (e.g., define some tests such as verifying that the banana is a fruit, but not all fruits are bananas);
8. Evaluation: based on validation and verification steps of the development phase from AMOD (e.g., perform the tests in the fruit ontology);
9. Post-game: the same as AMOD post-game, with complete ontology documentation (e.g., describe the ontology development method used and some commets that are useful for future reuse).

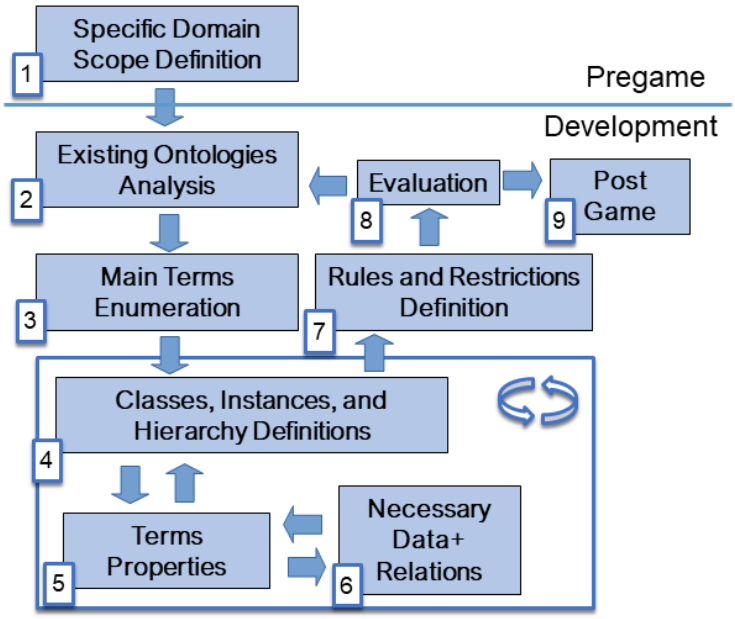

Figure 1: Manual ontology development based on 101 and AMOD methods.

3. The Semi-Automatic Way

As an initial disclaimer, it is essential to justify why we opted for a semi-automated instead of a fully automated way. As our goal is to enhance the specialist time, it is reasonable not to exclude this employee entirely from the process, which could trigger the fear of automation: by losing control of the process, it is possible that specialists may lose confidence in the automated tool, or they may fear that these machines could take over their job.

Therefore, we chose the semi-automatic way to balance manual labor reduction and control over the process by letting specialists perform refinement and validation steps and machines perform laborious, repetitive tasks. Some steps still need to be performed by humans, but the laborious steps are automated.

Our proposed semi-automatic scheme was based on guidelines found in the literature regarding automation of steps such as main terms enumeration, hierarchy definitions, terms properties definition, identification of necessary data and relations, and definition of rules and restrictions (Moraes S. M., 2012). We combine two specialized tools found in the literature: the first process Brazilian Portuguese texts (Hartmann, 2017), and the second provides automated ontology construction (Lamy, 2017). Additionally, we use an ontology editor to perform the manual process we use as a benchmark (Noy N. F., 2001b).

The complete semi-automatic scheme is illustrated in Fig. 2. It consists of six steps:

1. Syntactic Analyzer: we split the Brazilian Portuguese text provided into sentences, process each sentence, and obtain Resource Description Framework (RDF) triples in the format [subject, predicate, object]. For example, in the sentence “Apple is a kind of fruit”, the identified triple would be [apple, is a kind of, fruit];
2. Word Similarity: the candidate triples identified in the previous step are prioritized by calculating how frequently the subject and object words co-

occur to decide whether some specific triple should be prioritized or not. As an example, consider that the words “apple” and “fruit” occur in many Brazilian Portuguese texts. Therefore, the subject “apple” and the object “fruit” are expected to have a high value of similarity;

3. Selection and Adjustments: the business specialist must receive strong candidates triples in a spreadsheet format with five columns: subject, predicate, object, similarity, and hierarchical. The last column indicates whether the triple describes a hierarchical relationship or not and must be filled in by the specialist. In this step, the specialist must exclude and modify the triples according to his/her judgment (e.g., correct misspellings in subjects or objects of the triples). The triple [apple, is a kind of, fruit] is a hierarchical one, and another triple [person, eats, apple] is non-hierarchical;
4. Ontology Constructor: with the triples selected and corrected by the business specialist in the previous step, the tool automatically builds the ontology by identifying classes, subclasses, instances, properties, and relationships. In the example, the triple [apple, is a kind of, fruit] will result in an ontology with fruit class and apple instance, in a structure that resembles a mind map, and has the benefit of being understandable by humans and readable by machines;
5. Fine-tuning: the resulting initial ontology in this step can be refined and visualized in a specific program for ontology editing tools. The specialist must identify missing information and inconsistencies so that the natural language text can be modified and extended as required;
6. Validation: after specialist fine-tuning, another specialist known as a knowledge engineer validates the ontology.

The main contributions of our approach are (1) support for non-hierarchical relationships (e.g., the documents required for account opening), (2) support for Brazilian Protuguese and (3) Banking business domain.

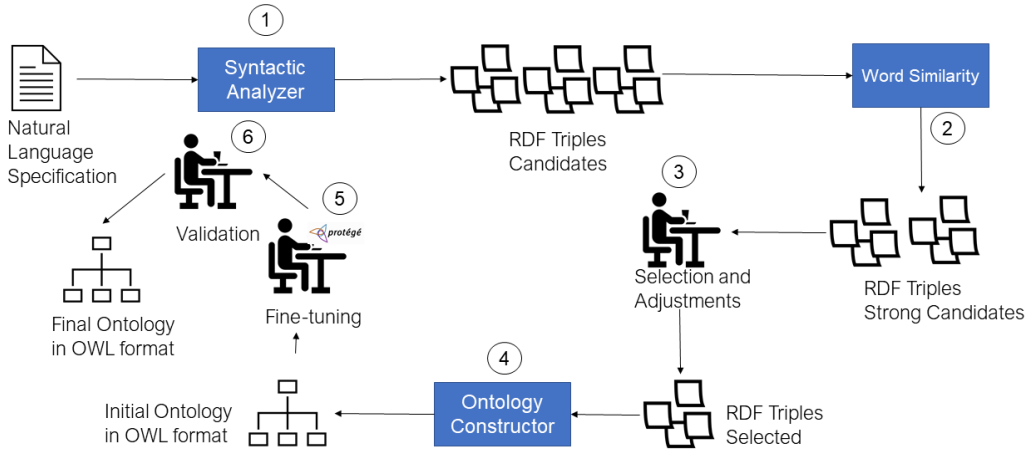

Figure 2: Proposed semi-automatic ontology development method

4. Some Examples

In this section, three examples of the banking domain are presented to showcase the proposed method capability to generate useful ontologies for a specific business domain. All the three ontologies described in the use cases were developed by the authors using the proposed semi-automatic scheme. They are based on Brazilian Protuguese texts written by business specialists.

4.1. Account Opening

It currently is possible to open a banking account using the personal mobile device, as it is not necessary to rely solely on physical agencies. However, in this digitalized alternative, a customer may have doubts regarding the documents required to open a banking account. This customer query may be answered by the customer service chatbot, which must use specific banking knowledge modeled in an ontology that we will create in this example.

The text which describes the account opening process is (free translation from Brazilian Portuguese to English): "*The solution will describe the account opening process. In the account opening, the desired bank account type is informed, and some documents must be provided. These account types may be: checking account, savings*

account, salary account, and international account. The necessary documents are RG, CPF, residence proof and income proof."

The original text in Brazilian Portuguese is: "*A solução descreverá o procedimento de abertura de conta. Durante a abertura de conta, informa-se o tipo de conta desejado e envia-se alguns documentos. Os tipos de conta podem ser: conta-corrente, conta-poupança, conta-salário e conta-internacional. Os documentos necessários são: RG, CPF, Comprovante de endereço, Comprovante de renda*".

The text is transformed into an ontology by performing the following steps:

1) In total, 11 RDF triples have been identified in the first step;
2) Word Similarity prioritizes the 11 triples in the second step;
3) The specialist selects and adjusts the triples provided in a spreadsheet. Verbs, subjects, and objects names were standardized, and each triple was classified as hierarchical or not. For example, the sentence "*These account types may be: checking account, savings account, salary account, and international account*" generates the triples [account types, maybe, checking account], [account types, may be, savings account], [account types, maybe, salary account], [account types, may be, international account]. The specialist can adjust the triples to [account, is, checking account], [account, is, savings account], [account, is, salary account], [account, is, international account], and classify all triples as hierarchical. Another example: from the sentence "*In the account opening, the desired bank account type is informed and some documents must be provided*", one generated triple is [account opening, must be provided, some documents], the specialist can modify it to [account opening, must provide, document], and classify it as non-hierarchical.
4) The ontology presented in Fig. 3 is generated. The specialist and knowledge engineer can visualize, adjust, and validate the ontology in the spelialized editing tool.

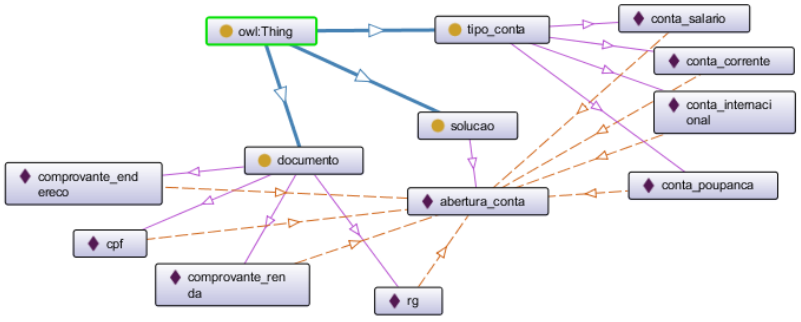

Figure 3: Account opening ontology created with with the semi-automatic method.

4.2. Investment Queries

The second example depicts Brazilian investments ontology construction from a natural language text. Consider the scenario where a customer wants to invest in some banking products but he/she is not sure about the different alternatives and taxes associated.

It is possible to use our semi-automatic scheme to integrate two ontologies from two different texts. The first text describes Brazilian investments at a top-level, with various fixed and variable income investments. The second text represents one of these in detail, namely the CDB (a fixed-income investment in Brazil). The specialist and knowledge engineer can use the proposed scheme to develop first a specific ontology for CDB investment, as illustrated in the ontology visualization of Fig. 4.

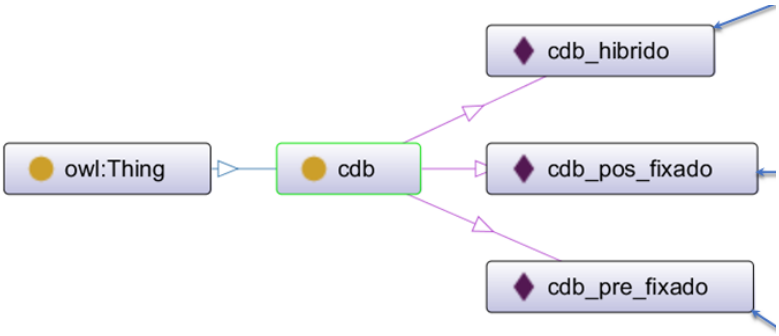

Figure 4: Ontology developed with specific CDB investment text.

The text that describes the investments at a top-level can be translated as follows: "*Investments can be divided into fixed income and variable income. These investments are analyzed in terms of investment risk, profitability, grace period, maturity, minimum investment, liquidity, taxes and administration fee. The main fixed income options are: savings, Direct Treasury (Selic Treasury, IPCA Treasury, Prefixed Treasury), CDB (Bank Deposit Certificate), LCI (Real Estate Credit Bill), LCA (Agribusiness Credit Bill), LIG (Guaranteed Real Estate Bill), LC (Letter of Exchange), LF (Financial Bill), debentures, debentures with incentives (without income tax), investment funds, COE (Certificate of Structured Transactions). The main variable income options are: stocks, equity funds, multimarket funds, real estate funds, derivatives, commodities and COE (Structured Transactions Certificate)*".

The original text in Brazilian Portuguese is: "*As aplicações podem ser de renda fixa e renda variável. Os principais pontos a analisar para escolher os melhores investimentos são: risco da aplicação, rentabilidade, prazo do investimento, custos e*

taxas, impostos. As principais opções de renda fixa são: poupança, Tesouro Direto (Tesouro Selic, Tesouro IPCA, Tesouro Prefixado), CDB (Certificado de Depósito Bancário), LCI (Letra de Crédito Imobiliário), LCA (Letra de Crédito do Agronegócio), LIG (Letra Imobiliária Garantida), LC (Letra de Câmbio), LF (Letra Financeira), debêntures, debêntures incentivadas (sem Imposto de Renda), fundos de investimento, COE (Certificado de Operações Estruturadas). Renda variável são aquelas cujo rendimento é imprevisível e depende de outros fatores, como a saúde da economia, por exemplo. As principais opções de renda variável são: ações, fundos de ações, fundos multimercado, fundos imobiliários, derivativos, commodities e COE (Certificado de Operações Estruturadas)".

The text that specifically describes the CDB investment has the following free translation: "*In the CDB, the amount invested, the liquidity of the security and the yield period must be defined. There are three types of CDB: pre-fixed, post-fixed and hybrid. The CDB matures in 2 years, with a 0.3% administrative fee, high liquidity, 1 month grace period and low risk, with two types of taxation: income tax and IOF. The minimum investment for CDB starts from R$ 10000. The floating CDB has 100% of the CDI yield, the fixed CDB has 120% of the CDI and the hybrid has 110% of the CDI*".

Therefore, the first text provided general investments properties, and the second text provided specific CDB information. The idea is to populate relationships given as common to all investments in the top-level text, such as grace period, minimum investment, and taxes, with the information given by the second text.

The resulting ontology (Fig. 6) was derived from the information extracted according to the triples generated by the proposed semi-automatic process (following the same steps described earlier), integrating the ontologies generated by the two texts. The investments were divided into two main categories (fixed and variable income), and the process could assimilate the mentioned examples. Then, the CDB category was specified by the CDB types (pre-fixed, post-fixed, and hybrid) provided by the specific text.

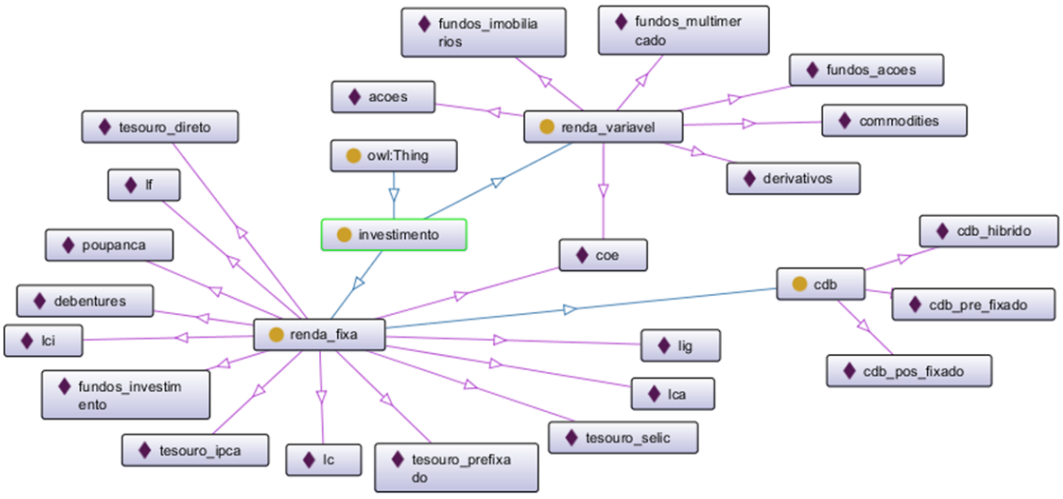

Figure 6: Ontology developed combining general and specific investment texts.

4.3. Financial Transactions

The third use case depicts the task ontology development for balance inquiry, transfer, and payments operations, which are everyday financial transactions performed by thousands of banking customers daily. It was constructed using the manual approach described in Section 2 and the semi-automatic process presented in Section 3.

The free translation of the text used in this use case is: "*The procedures covered in the solution should be payments, transfers, and balance verification. during these procedures, the person must be authenticated with the bank branch, account, and 8-digit password. Here, a person can be understood as an individual or a legal entity. while the natural person is a favored individual or sending individual, the legal person is an abstract subject such as companies, political parties, associations, among other beneficiary organizations, and senders. The banks of the persons involved must be recognized as the sender's bank and the beneficiary's bank. The bank branch, being differentiated as the sender's branch or the beneficiary's branch, is the part of the bank that offers customers personal and automated services. Branch managers are people specialized in banking matters who stay at these branches to*

serve clients. The types of accounts are checking accounts, salary accounts, or savings accounts, in the case of the sender's account or the payee's account. To verify the balance, the person is required to be authenticated with the number and type of bank account (checking account, savings account, or salary account) and the branch. the payments described are deposit, bank slip, and card. to make deposits, the following is required: the name of the person, the number of the CPF or CNPJ, the bank number, the branch, and the beneficiary's account. to pay a bill, you need the Bill code, the effective date of payment, and the sender's account number and type. to pay a card bill, the sender's account number and type, the bill to be paid, the amount of this payment, as well as its effective date, are required. to carry out transfers, some data about the recipient is required, such as the name or bank code, branch number, account number and type, payee's CPF or CNPJ, payee's full name or corporate name, amount to be transferred and the effective date. in addition to the payee's data, the transfer requires the sender's branch, account, and bank. transfer options are intrabank transfer (those carried out between people who have an account in the same bank) and interbank transfer (procedure between different banks). the interbank transfer, in turn, can be carried out via TED or DOC."

The original text in Brazilian Protuguese is: "*O documento que descreve o conhecimento que a solução deve ter, está destrinchado em torno de procedimentos bancários pertencentes ao escopo de trabalho definido. os procedimentos abordados são: pagamentos, transferências e verificação de saldo. durante esses procedimentos, a pessoa precisa estar autenticada com a agência bancária, conta e a senha de 8 dígitos. Aqui, pessoa pode ser entendido como uma pessoa física ou pessoa jurídica. enquanto a pessoa física é um indivíduo favorecido ou indivíduo remetente, a pessoa jurídica é um sujeito abstrato como empresas, partidos políticos, associações, entre outras organizações beneficiárias e remetentes. Os bancos das pessoas envolvidas devem ser reconhecidos como: banco do remetente e banco do beneficiário. A agência bancária, sendo diferenciada como a agência do remetente ou a agência do favorecido, é a parte do banco a qual oferece aos clientes atendimentos pessoais e automatizados. Os gerentes de agência são pessoas especializadas em assuntos bancários que ficam nestas agências para atender os*

clientes. já os tipos de contas são: conta-corrente, conta-salário ou conta-poupança, trantando-se da conta do remetente ou da conta do favorecido. para a verificação de saldo demanda-se que a pessoa esteja autenticada com o número e tipo de conta bancária (conta corrente, conta poupança ou conta salário) e a agência. já os pagamentos descritos são depósito, boleto e cartão. para realizar depósitos, demanda-se: o nome da pessoa, o número do cpf ou cnpj, o número do banco, a agência e a conta do beneficiário. para pagar um boleto, é necessário o código do boleto, a data de efetivação do pagamento e o número e tipo de conta do remetente. para o pagamento de fatura de cartão, é requisitado o número e tipo da conta do remetente, a fatura a ser paga, o valor deste pagamento, assim como sua data de efetivação. para a realização de transferências, são necessários alguns dados sobre o destinatário como: o nome ou código do banco, número da agência, número e tipo da conta, cpf ou cnpj do favorecido, nome completo ou razão social do favorecido, valor a ser transferido e a data de efetivação. além dos dados do favorecido, para a transferência é preciso ter a agência, a conta e o banco do remetente. têm-se como opções de transferências: a transferência intrabancária (aquelas realizadas entre pessoas que possuem conta no mesmo banco) e a transferência interbancária (procedimento entre bancos diferentes). a transferência interbancária, por sua vez, pode ser realizada via TED or DOC".

The comparison shows that the task ontology constructed with the proposed semi-automatic approach resembles the manual ontology. There are five blocks:

- Agency and bank classes, and related instances;
- Account class and its instances;
- Person class, natural person and legal entity subclasses, and related instances;
- Payment process information;
- Balance inquiry and transfer procedures information.

Some differences regard the legal entity subclass, whose instances can be found only in the semi-automatic ontology. Account types are instances in the semi-automatic ontology, while account types are classes in the manual ontology.

As there is no unique way to develop a domain according to methodology 101 premises, this information is not wrong. Still, a given configuration may be more

interesting than the other, depending on the usage. The text of the experiment was designed to describe the data needed to carry out different banking transactions, and the generated ontology captured this same information, as can be seen in Fig. 7. We thus conclude that the semi-automatic process did not lose necessary data and provided acceptable results.

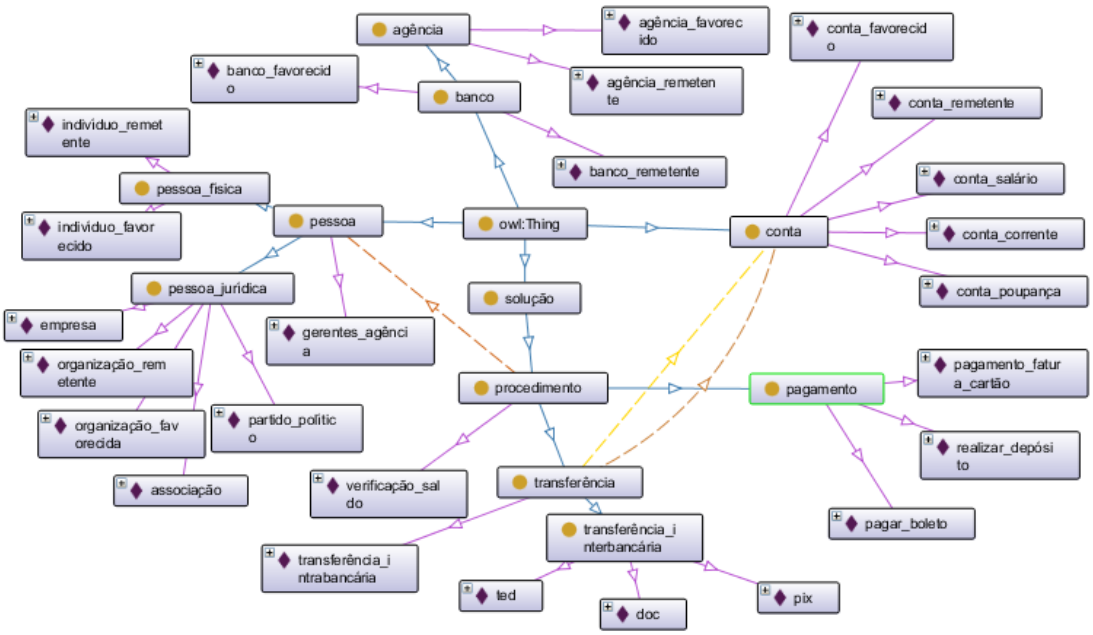

Figure 7: Semi-automatic ontology developed for Financial Transactions.

Considering the third use case, an empirical time cost was estimated for each approach, as depicted in Table 1.

Table 1: Time Effort Comparison (time in hours)

Step	**Manual**	**Semi-automatic**
1. Specific Domain Scope Definition	2	2
2. Existing Ontologies Analysis	2	2
3. Main Terms Definition	5	2
4. Hierarchy Definitions	4	1
5. Terms Properties	6	1
6. Necessary Data + Relations	6	2
7. Rules and Restrictions Definition	2	2
8. Evaluation	5	5
9. Post Game	5	5
Total	**37**	**22**

The semi-automatic approach could reduce the total time needed from 37 hours to 22 hours, a relative reduction of 40% (15 hours/37hours) by improving 4 out of 9 steps.

5. Related Work

Some semi-automatic proposals for information retrieval from texts are present in the literature. However, most approaches do not support non-hierarchical relationships or Brazilian Portuguese, and they are not specialized in the banking domain.

One approach starts with a small ontology constructed by domain specialists manually. The authors expand this ontology with taxonomic and non-taxonomic relations of existing concepts found in the WordNet knowledge base and validate its time reduction in a medical domain use case (Zhou, 2006). Another approach proposes semi-automatic ontology development from an insurance company intranet texts (Kietz, 2000). Both related works are based on English texts. As an alternative to our semantic analyzer, the Formal Concept Analysis method obtains semantic information from English texts and uses this information to build ontological structures (Moraes S. a., 2012).

It is also possible to use artificial intelligente to obtain information from texts (Liu, 2018). In this related work, a machine learning model learns the mapping from natural language text in English to information in RDF triples format (i.e., subject-predicate-object). However, this method could only identify one triple by each sentence, while our approach can identify more than one triple in each input sentence. That is, our approach can obtain more information by each sentence found in the text provided.

A survey regarding semi-automatic ontology development tools for Brazilian Portuguese presented mechanisms that could identify hierarchical relationships, such as “apple is a kind of fruit”, but could not identify non-hierarchical relationships, such as “person may eat fruits” (Zahra F. M., 2014). For example, PORONTO, a solution

customized to the health domain that supports Brazilian Portuguese, identifies only taxonomic relationships (Zahra F. M., 2013).

To the best of the authors' knowledge, our solution is a pioneer in retrieving information from Brazilian Portuguese texts for the banking domain in a semi-automatic way. Existing solutions do not support Brazilian Portuguese and Banking, as they are not customized for this specific business area. We consider that our method may be useful for other languages and specific areas, which are opportunities that we can explore in future work.

6. Final Considerations

We presented a semi-automatic ontology development approach based on natural language descriptions in Brazilian Portuguese to optimize the laborious manual process of ontology development for banking customer service chatbots.

Our method integrated natural language processing tools to retrieve Brazilian Portuguese texts and model information in ontology format with ontology generation tools. The banking use cases and the comparison with manual process indicated the proposed method's capability to optimize specialist time effort, as initially proposed.

Compared to related work found in the literature, our proposal's novelty regards the support for non-hierarchical and hierarchical relationships, support for Brazilian Portuguese, and specialization for the banking domain. Our approach further extends the state-of-the-art by identifying more information in each sentence. We contribute the automation of knowledge based intelligent systems by reducing the level of intervention required, as the thinking is transferable across different disciplines.

In future work, it is possible to automate the specialist refinement and validation steps. Another research opportunity is to extend our tool to support a bottom-up approach so that a large banking ontology could be constructed incrementally. This

scheme could lessen knowledge management effort, as each specific ontology could be constructed individually and integrated into a top-level ontology.

7. Acknowledgments

The authors thank the non-anonymous reviewers Reginaldo Arakaki, Professor of Software Engineering at the Polytechnic School of the University of São Paulo (USP), and Renato Manzan from Microsoft for their valuable comments on the initial intra-disciplinary paper. The authors acknowledge the beta-readers Rony Sakuragui and Ana M. B. Barufi from Bradesco for their valuable contributions as beta-readers for this trans-disciplinary communication, and M. Cristina V. Borba, an experienced Portuguese-English academic translator, for the English proof-reading.

The authors also thank the support provided by the Foundation of Support to the University of São Paulo (FUSP) and the Laboratory of Computer Architecture and Networks (LARC) at the Computing and Digital Systems Engineering Department (PCS), Polytechnic School, University of São Paulo, Brazil.

References

Abdelghany, A. S. (2019). An agile methodology for ontology development. *International Journal of Intelligent Engineering and Systems*.

Bennett, M. (2013). The financial industry business ontology: Best practice for big data. *Journal of Banking Regulation*, pp. 255-268.

Bimba, A. T.-H. (2016). Towards knowledge modeling and manipulation technologies: A survey. *International Journal of Information Management*.

Debenham, J. (2012). *Knowledge Engineering: Unifying Knowledge Base and Database Design.* pringer Science & Business Media.

Dudycz, H. a. (2015). Conceptual design of financial ontology. *2015 Federated Conference on Computer Science and Information Systems (FedCSIS).* IEEE.

Gangemi, A. a. (2006). Modelling ontology evaluation and validation. *European Semantic Web Conference.* Springer.

Guarino, N. (1997). Understanding, building and using ontologies. *International journal of human-computer studies*, pp. 293-310.

Happel, H.-J. a. (2006). KOntoR: an ontology-enabled approach to software reuse. *Proc. Of The 18Th Int. Conf. On Software Engineering And Knowledge Engineering.*

Hartmann, N. a. (2017). Portuguese word embeddings: Evaluating on word analogies and natural language tasks. *arXiv preprint arXiv:1708.06025*.

Jones, D. a.-C. (1998). Methodologies for ontology development. *University of Liverpool website*.

Kendal, S. L. (2007). *An introduction to knowledge engineering.* Springer.

Kietz, J.-U. a. (2000). A method for semi-automatic ontology acquisition from a corporate intranet. *EKAW-2000 Workshop "Ontologies and Text"*, (pp. 1-14). Juan-Les-Pins, France.
Lamy, J.-B. (2017). Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies. *Artificial intelligence in medicine*.
Liu, Y. a. (2018). Seq2rdf: An end-to-end application for deriving triples from natural language text. *arXiv preprint arXiv:1807.01763*.
Moraes, S. a. (2012). Combining Formal Concept Analysis and semantic information for building ontological structures from texts: an exploratory study. *LREC.*
Moraes, S. M. (2012). Construçâo de Estruturas Ontológicas a partir de textos: um estudo baseado no método Formal Concept Analysis e em papéis semânticos.
Noy, N. F. (2001a). *Ontology development 101: A guide to creating your first ontology.* Stanford knowledge systems laboratory technical report KSL-01-05.
Noy, N. F. (2001b). Creating semantic web contents with protege-2000. *IEEE intelligent systems*.
Shaw, M. L. (1990). Modeling expert knowledge. *Knowledge Acquisition*.
Tang, H. a. (2011). Ontologies in financial services: Design and applications. *2011 International Conference on Business Management and Electronic Information.* IEEE.
Tecuci, G. a. (2016). *Knowledge engineering: building cognitive assistants for evidence-based reasoning.* Cambridge University Press.
Tudorache, T. a. (2008). Supporting collaborative ontology development in Protégé. *International Semantic Web Conference.* Springer.
Van Heijst, G. a. (1997). Using explicit ontologies in KBS development. *International journal of human-computer studies*, pp. 183-292.
Wouters, B. a. (2000). The use of ontologies as a backbone for use case management. *European Conference on Object-Oriented Programming (ECOOP 2000), Workshop: Objects and Classifications, a natural convergence.*
Zahra, F. M. (2013). Poronto: ferramenta para construção semi automática de ontologias em português. *Journal of Health Informatics*.
Zahra, F. M. (2014). Ferramentas para aprendizagem de ontologias a partir de textos. *Perspectivas em Ciência da Informação*.
Zhou, W. a.-l. (2006). A semi-automatic ontology learning based on wordnet and event-based natural language processing. *2006 International Conference on Information and Automation.* IEEE.