

Anaphora Resolution in Business Process Requirement Engineering

Riad Sonbol¹, Ghaida Rebdawi², Nada Ghneim³

^{1,2}Department of Informatics, The Higher Institute for Applied Science and Technology, Syria

³Faculty of Informatics, Damascus University, Syria

Article Info

Article history:

Received Oct 15, 2017

Revised Jan 12, 2018

Accepted Jan 20, 2018

Keyword:

Anaphora resolution

Business process modeling

Natural language processing

Process model discovery

Requirement engineering

ABSTRACT

Anaphora resolution (AR) is one of the most important tasks in natural language processing which focuses on the problem of resolving what a pronoun, or a noun phrase refers to. Moreover, AR plays an essential role when dealing with business process textual description, either when trying to discover the process model from the text, or when validating an existing model. It helps these systems in discovering the core components in any process model (actors and objects). In this paper, we propose a domain specific AR system. The approach starts by automatically generating the concept map of the text, then the system uses this map to resolve references using the syntactic and semantic relations in the concept map. The approach outperforms the state-of-the-art performance in the domain of business process texts with more than 73% accuracy. In addition, this approach could be easily adopted to resolve references in other domains.

Copyright © 2018 Institute of Advanced Engineering and Science.

All rights reserved.

Corresponding Author:

Riad Sonbol,

Informatics Department,

The Higher Institute for Applied Science and Technology,

Damascus, Syria.

Email: riad.sonbol@hiast.edu.sy

1. INTRODUCTION

Anaphora are backward references to expressions (usually nouns) which have previously been mentioned in the text such as "they" and "it" in the sentence "**They** buy the issue, then resell **it** to the public." According to its locations [1], anaphora could be intra-sentential when the reference is located in the same sentence like "it" in the last example, or could be inter-sentential when it is located in previous sentences like "they" in the last example. Another way to categorize anaphora is based on its form [2], the most common type according to this way of categorization is pronominal anaphora which could be personal (e.g. he, she, it, his), demonstrative (eg. this, that, those), or reflexive (e.g. himself, itself). Other types are quantifiers (e.g. one, each, some), ordinal (first, former, latter), wh-anaphora (e.g. who, which), in addition to the Pleonastic it, and idiomatic forms.

The process of identifying anaphoric relations is called *anaphora resolution (AR)*. AR is one of the main tasks in Natural Language Processing (NLP) [3]. It plays an important role in many applications [4]-[7] including Information Extraction, Text Summarization, Question Answering, Machine Translation, Opinion Mining, Requirement analysis, and discovering models from textual description which is the domain we are focusing on in this paper. However, AR is considered as one of the main challenging problem in NLP because of its high-dependency on both syntactic and semantic processing; the accuracy of these two types of pre-processing is still far from ideal, which add more difficulties on building an accurate AR system [8]. Many researchers have worked on building an AR system from different perspectives; part of them used rule-based approaches, and others followed a statistical procedure based on tagged corpora. In rule-based approaches, [9], [10], a set of manual-encoded rules are used. These rules are based on various linguistic

features (mainly morpho-syntactic features such as part-of-speech and gender/number information). The best-achieved performance in MUC-7 was around 70% precision with 60% recall, which is still lower than what we need in practical applications. These rules are limited in general and does not work equally in different domains.

Statistical approaches apply Machine Learning algorithms, such as Decision Trees and Support Vector Machines [11]-[13]. Usually, these learning algorithms are applied on lexical, syntactic, semantic, and knowledge-based features. In general, the achieved performance is close to the best-performing rule-based system in MUC-6 and MUC-7. Other works present hybrid approaches combining rule-based and machine-learning components [14], which aim to combine the advantage of each of these approaches.

Since the problem of AR is still far from being solved [15], many papers focused on domain related AR system to support a specific domain or a specific application. Many papers have been proposed to resolve anaphoric NPs in biomedical texts [16]-[18], machine translation [19], or busines process domains [20], [21], [7]. In the next sections, we will focus on the domain of process modeling and the importance of AR systems in this domain, then we will review some related works in sections 3. Section 4 presents our proposed approach, and then we evaluate this approach with a comparison to other researches in Section 5. We conclude our paper in Section 6.

2. BUSINESS PROCESS MODELING AND THE NEED FOR AR SYSTEMS

Modeling is one of the core tasks in business process management (BPM) [22]. It aims to create representations (usually called models) of the processes of an organization to understand them, documenting their details, analyzing their performance to determine opportunities for improvements, or representing the target process state.

Due to its challenges, modeling represents the most critical step in BPM life cycle. It is the most time consuming and costly task; it requires conducting a number of meetings, workshops, interviews between modeling experts and process performers to acquire the required knowledge in a highly interactive and repetitive approach. According to [23], building as-is model consumes about 60% of the overall time spent in a workflow project.

On the other hand, in most organizations, the required information are available in textual forms; 85% of the information in companies are stored in unstructured documents mostly textual. This includes polices, reports, forms, manuals, knowledge management systems, and emails messages [24]. In addition, most process performers are accustomed to expressing their needs in natural language [25]. All these textual information represent a potential sources of knowledge needed in building the model.

These facts raised the question of how researchers can save this cost by building tools that could support modeling experts in their manual workload. These systems will not replace modeling expert but will help them in creating models more efficiently in terms of the required time, cost, and quality [26]. According to [7], substantial savings are possible by providing such automation tools.

Natural Language Processing (NLP) plays an essential role in extracting the model from documents, since one need to handle many complex challenges including extracting activities, ordering them, and dealing with concurrency, and loops [27]. Anaphora resolution is one of the problems that must be tackled when analyzing textual process descriptions, as many concepts, tasks, actors and relationship between phrases would be missed if the references in text are not resolved. For instance, the word “it” in the sentence (“If it is not available, it is back-ordered”), could not be meaningful unless we consider the previous context. Moreover, some activities can be split up into more than one sentence and anaphora analysis helps to connect and recognize these sentences as one activity. The next example shows the importance of good AR system when discovering the model from texts:

- a) Finally, **she** assigns the order to the waiter. (*we need AR to discover the Actor*)
- b) Once the food, juice, and cart are ready, the waiter delivers **it** to the guest's room. (*we need AR to detect the Object*)
- c) Usually **they** play a triple role: First, **they** provide the company with procedural and financial advice, then **they** buy the issue, and finally **they** resell **it** to the public. (*We need AR to detect both the object and actors*)
- d) If Service Management is sure they can analyze it, they perform the analysis. (*We need AR to detect the condition*)

In the case of business process modeling, the most common usage of anaphora is to refer to an actor, object or resource. These three kinds of noun phrases represent the most frequent entities in any textual description. This fact adds more difficulties to AR systems in this domain. Actually, using simple distance heuristics would not be sufficient to get an accepted accuracy, most AR systems in this domain use syntactic features to improve the accuracy.

3. RELATED WORKS

In [20], an AR system has been proposed to support the similarity check between an activity in the process model and a sentence in the natural language text. This approach uses Stanford Parser to retrieve a dependency tree. This tree would be used to identify objects in the sentences based on the relations of direct objects and nominal subjects. If the sentence where the Anaphora occurred has no objects, the anaphoric is replaced by an object present in the previous sentence.

[21] Introduces three different anaphora resolution approaches: a simple lexical approach and two further approaches based on association rules which were created during a statistical analysis of a workflow corpus using a sequential pattern mining that was presented by Agrawal [31]. The paper evaluates these three approaches using 37 workflows created by a human expert, all of them are from cooking domain, the best achieved precision was around 51% with 30% recall.

[7] Introduces a heuristics-based approach to resolve determiner and pronouns. The approach uses the distance between anaphora and nouns, in addition to a set of syntactic matching features. Using 47 process descriptions covering various domains, the authors evaluate their work and compare their approach with two generic anaphora resolution system (BART [28] and Reconcile [29]). The results show that their application-oriented approach achieved an accuracy of 63.06% while both generic approaches were not able to identify more than 42% of the cases correctly.

4. OUR APPROACH

To resolve an anaphora, our approach is based on two main stages: First, we process the whole text to extract its concepts map, i.e. its concepts and the semantic relationships between these concepts, then in the second stage we use the extracted map to resolve anaphora to the most similar concept based on a similarity measure. This similarity measure would have two important properties: (1) it would be based on the relationships in the map which represents the whole text i.e. not only based on a few sentences before or after the anaphora. (2) it is based on a domain-specific semantic network, not on a generic network such as wordNet, this fact make the measurements closer to the context of the concepts in the text. In the next paragraphs, we will give more details about each of these two stages:

3.1. Initial concept map generation

1) Morphological and Lexical Analysis

In this step, we split the text into sentences and sentences into words. Then, we apply Stanford Lemmatizer on each word to remove inflectional endings and to find the base form of the word (i.e. lemma).

2) Syntactic Analysis

First, we use Stanford pos-tagger [30] to tag each word with its suitable POS tag which indicates the syntactic role of the word (such as plural Noun, singular Noun, Adverb, Adjective, etc.). This syntactic information would help us to determine the entities and the concepts in next semantic processing steps. Then, we generate the dependency tree for each sentence using the work of Chen and Manning [31]. This parser concludes the dependency tree depending on neural network by representing all words, POS tags and dependency relationships' labels as dense vectors. According to the authors, the parser is able to parse more than 1000 sentences per second at 92.2% unlabeled attachment score on the English Penn Treebank, which make it suitable for our needs in this work.

3) Noun Phrase Boundary (NPBT) Tagging

In this step, we used a rule-based tagging approach to tag words with three possible Noun Phrase Boundary (NPB) Tags, these three tags would be used later to determine the boundaries of the noun phrases:

- a. "SE" (Start of Entity): the tagged word is the first word in a noun phrase.
- b. "IE" (In Entity): the tagged word is part of a noun phrase (but should not be the first word in it).
- c. "O" (Out): the tagged word could not be part of a noun phrase.

Our system uses 18 rules to tag the words with these three NPB tags. The left side of each rule represents its contextual conditions. We describe the context by the morphological, syntactical tags for the related words and the last three mentioned noun phrase boundaries' tags. When any sequence of words in the text satisfy the contextual condition of a rule, we apply the set of changes in noun phrase boundaries' tags defined in the right side of the rule.

Example:

In the following rule:

```
"Lemma=of@0# NPBT=O!@-1# NPBT =O!@1
=> NPBT =IE@0"
```

the contextual condition of this rule is (Lemma=of@0# NPBT =O!@-1# NPBT =O!@1) which means:

- a. Lemma=of@0: the lemma of the word in the position zero is "of".
- b. NPBT =O!@-1: the NPB tag is not O (! means a negative condition) in the previous word (since @-1 means "at the index -1")
- c. NPBT =O!@1: the NPB tag is not O in the next word (since @+1 mean "at the index +1")

The result would be tagging the word at position 0 in the target context with the tag (IE).

4) Concepts Detection

In this step, we scan the text to extract the noun phrases using the result of NPB-tagger. Any sequence words with tags SE and IE represents a possible concept. Anaphora cases such as "He", or "She" are considered as possible concepts in the last tagging step. To distinguish between the occurrence of the same pronoun in different sentences we add a location suffix to each anaphora when extracting it as a possible concept in this stage. For example, if we have the pronoun "he" as a third word in the second sentence we add the concept he_2_3 where 2_3 is the location prefix. In this way, each pronoun would be stored as a distinct concept even if more than one pronoun refer to the same named entity. Later, we will resolve these anaphora cases and merge such cases.

5) Alias Detection

Some concepts could appear in different forms in the text. In this step, we merge all of these possible concepts in one concept. To do so, we detect the possible merges by checking if any of the possible concepts is an ending for another possible concept (regardless of the morphological changes). For example, "response comment" and "the comment" represent a possible merge since the second concept is an ending for the first one.

Practically, we found that this heuristic is sufficient to detect the possible merges, but rarely there are cases where users use completely different words (synonyms) to express the same concept when describing a business process. When we find two possible merges such as the case of "document" in ["document", "confirmation document"], ["document", "rejection document"], we choose the nearest concept to merge with, in terms of number of words between each of these two concepts.

After merging concepts, we consider the longest string (in terms of number of words) as a main concept, and we consider all other strings as aliases. The main concept would be used when building the model whenever any of the aliases were used. As notes, this step does not cover detecting anaphora aliases since these anaphoras have not been resolved yet.

6) Relationship Generation:

After extracting the concepts from the text, we need to connect them by semantically relationship. To do that we add four kinds of relationships between concepts:

- a. If a concept X starts with a concept Y, we add a relationship from X to Y, for example: "A Confirmation" and "Confirmation Document", the label of this relationship is "related to".
- b. By checking the dependency tree (the output of syntactic analysis), If there is a verb connecting two concepts using subject and object dependency relationships, we create a relation between them labeled by the verb. We consider a verb is connecting two concepts if it connects any word from the first concept with any word from the second one. Here we differ between two types of connections: the verb has a relation to two different concepts, there is a relation from the first concept to the verb, and a relation from this verb to another concept.

In both cases, our processing deal with conjunction case such as: X can accept or reject Y", "X process Y and Z.", "X and Y Process X".

- c. If there is a dependency relationship between any two concepts, we reflect it as a relationship in the concept map, we label this relationship by the name of the dependency + the specified string if exist.

3.2. Anaphora resolver

We resolve anaphora using the concepts and the relationships which have been extracted till now. Figure 1 represents the result of applying the last steps on the following process: *"First, the Manager checks the open leads. Afterwards, he selects the top five ones. He then tells his Sales Assistant to call the contact person of the leads. The Sales Assistant calls each customer. If someone is interested, he sends a note to the Manager. The Manager then processes the lead. Otherwise, he phones the next customer."*

The last concept map contains 6 anaphoras: he_2, he_3, he_5, he_7, ones_2, and someone_5. The attached suffix (such as _2 and _7) define anaphora's location in the text (he_2 refers to the "he" occurring in the second sentence, while he_7 refers to its occurrence in sentence 7). For each anaphora, we consider all previous concepts, in terms of their occurrence in the text, as a possible resolver. Our objective in this step would be ranking these possible concepts and choosing the most suitable one. We do that by comparing the context of the anaphora and the context of each possible concept. The context could be achieved by considering concepts' position in the concept map i.e. its in-relationships (relations from other nodes), and out-relationships (relations to other nodes). For example, the context of "he_7" could be defined by its

relation with "customer" node: it is the one who "phones" the "customer". In the same way, "he-5" is a concept which "send" a "note", and the one who "send sth to" the "manager".

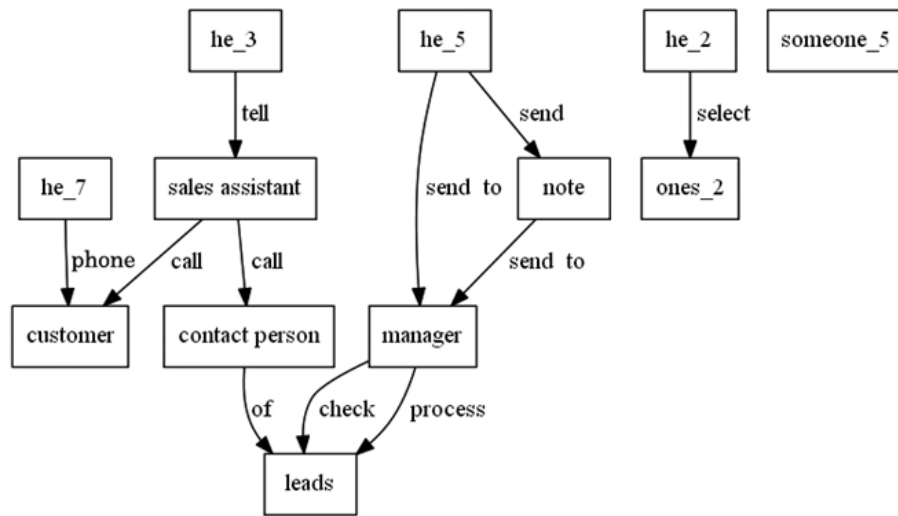


Figure 1. sample output of applying initial concept map generation stage

We give each possible concept a score for being a resolver for the anaphora in the following way:

$$Score_{anaphora} (Cocept) = \max_{\substack{rel1 \in concept.relaions \\ rel2 \in anaphora.relations}} sim(rel1, rel2)$$

i.e. we calculate the similarity between each relationship connected with the anaphora in concept map, and each relation connected with the concept we want to rank. Then the score of this concept would be the max score we find between any two relations.

The similarity between two relationships in the concept map would be calculated as following:

- a. If *rel1* is an in-relation for the concept and *rel2* is an out-relation for the anaphora
- b.

$$sim(rel1, rel2) = 0$$

Since these two relations are no compatible in term of direction.

- c. If *rel1* is an out-relation for the concept and *rel2* is an in-relation for the anaphora

$$sim(rel1, rel2) = 0$$

Since these two relations are no compatible in term of direction.

- d. If *rel1* is an in-relation for the concept and *rel2* is an in-relation for the anaphora

$$sim(rel1, rel2) = WuP_Sim(rel1.label, rel2.source) \times WuP_Sim(rel1.source, rel2.source)$$

since both relations are in-relations, we should consider the source and the label of each of them.

Wu_P simalrity (or Wu-Palmer similarity) is one of the common wordNet-based semantic similarity [32] which check the similarity between two strings, it gives similarity on a [0,1] scale reflecting the minimum distance between any two synsets of two given concepts in WordNet.

- e. If *rel1* is an out-relation for the concept and *rel2* is an out-relation for the anaphora

$$sim(rel1, rel2) = WuP_Sim(rel1.label, rel2.target) \times WuP_Sim(rel1.source, rel2.target)$$

since both relations are out-relations, we should consider the target and the label of each of them

After applying the last calculations, we choose the concept with max acore. When two concepts have the same score, we choose the closest one to the anaphora.

Exceptional Cases:

In addition to the previous calculations, we added these two heuristics:

- a. If there is a semantic relation between the anaphora and a possible concept, we exclude this concept from the possible resolver by giving it a negative probability.
- b. If there is a previous anaphora which equal the current anaphora lexically, we multiply its score by two.

5. RESULT AND DISCUSSION

To evaluate our work, we used the dataset collected by [7]. We decided to use this dataset since: (1) it is the best dataset we know in this domain in terms of its size and variety, as it consists of 47 textual process descriptions (in total, 432 sentences) collected from different sources (academic and industry) and from different domains (computer, hotels, manufacturing, HR...). (2) we can compare our results with the results of Friedrich AR which is designed specifically for this domain, as a part of one of the best-known text-to-model systems according to [26]. Moreover, we can compare our results with two generic AR systems: BART and the Reconcile.

Our approach was able to resolve more than 73% of anaphoras correctly, which outperform the results of Friedrich [7] which reached about 63% when using the same dataset. In addition, it clearly outperforms BART and Reconcile systems in the domain of business process requirement engineering: according to [7], these two generic AR systems were not able to correctly identify more than 42% in this dataset: BERT can resolve only 41.44% of the cases on the same dataset, while Reconcile can resolve 39.64% correctly. Of course, these results could not be generalized to other domains since BART and the Reconcile are generic AR systems. The following table summarizes the results:

Table 1. Accuracy of the four AR systems

	Our Approach	Friedrich	BART	Reconcile
# of ref	111	111	111	111
Resolved Correctly	82	70	46	44
Accuracy	73.87%	63.06%	41.44%	39.64%

The usage of the concept map was the key advantage in our approach comparing to other works. In many cases, anaphora could not be resolved in its narrow context, it could be resolved only when connecting the concepts with all related properties and all of its semantic relations in the whole text, the effect of this kind of processing become clearer when dealing with inter-sentential anaphora when the anaphora and the related resolver are located in two different sentences. Our statistics show that our approach gives correct results 1.25 times better than Friedrich approach in inter-sentential anaphora cases, while it gives the correct answer 1.08 times better in the case of intra-sentential.

Table 2. Comparing the accuracy between intra-sentential and inter-sentential anaphora

	Our Approach	Friedrich	Improvement
intra-sentential cases	82.2%	75.6%	1.08
inter-sentential cases	68.2%	54.5%	1.25

Table 2 show how accuracy of both systems (Friedrich and our approach) change depending on anaphora type, the first two columns show the accuracy of both systems in each type, and the last column calculate improvement percentage (example: $1.25=68.2/54.5$).

6. CONCLUSION

In this paper, we presented a domain specific anaphora resolution system, focusing on the domain of business process requirement engineering. The approach deals with the text as one block, and generates a concept map reflecting the concepts and the semantic relationships in the text. Based on this map, a semantic similarity measurement has been proposed to resolve references. The accuracy of our system outperforms well-known AR systems in this domain, using Friedrich dataset[7], we reached more than 73% accuracy outperforming similar approaches. These results encourage us to test this approach in more domains and to adopt the algorithm to be a domain-oriented anaphora resolution system; concept map generation should be tested in other domains to be sure that it gives good accuracy in all domains not only in the domain we focus on in our work.

REFERENCES

- [1] Iida R, Inui K, Matsumoto Y, “Zero-anaphora resolution by learning rich syntactic pattern features”, *ACM Transactions on Asian Language Information Processing (TALIP)*, 2007 Dec 1, vol. 6, no. 4, pp. 1.
- [2] Denber M, Automatic resolution of anaphora in English, *Eastman Kodak Co*, 1998 Jun 30.
- [3] Mitkov R, Anaphora resolution, *Routledge*, 2014 Feb 24.
- [4] Akkineni H, Lakshmi PV, Babu B, “Online Crowds Opinion-Mining it to Analyze Current Trend: A Review”, *International Journal of Electrical and Computer Engineering (IJECE)*, 2015 Oct 1, vol. 5, no. 5.
- [5] Khamayseh, Faisal T, “Inferring Student’s Chat Topic in Colloquial Arabic Text using Semantic Representation”, *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 4 (2016), 1897.
- [6] Kilicoglu H, Fiszman M, Demner-Fushman D, “Interpreting consumer health questions: The role of anaphora and ellipsis”, *In Proceedings of the 2013 Workshop on Biomedical Natural Language Processing 2013*. Aug 8, pp. 54-62
- [7] Friedrich F, Mendling J, Puhlmann F, “Process model generation from natural language text”, *In Advanced Information Systems Engineering 2011*, pp. 482-496, Springer Berlin/Heidelberg.
- [8] Poesio M, Stuckardt R, Versley Y, Anaphora Resolution, Springer, 2016.
- [9] Haghighi A, Klein D, “Simple coreference resolution with rich syntactic and semantic features”, *In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3 2009 Aug 6*, pp. 1152-1161.
- [10] Mitkov R “Factors in anaphora resolution: they are not the only things that matter: a case study based on two different approaches”, *In Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts 1997 Jul 11*, pp. 14-21.
- [11] Soon WM, Ng HT, Lim DC, “A machine learning approach to coreference resolution of noun phrases”, *Computational linguistics*, 2001 Dec, vol. 27, no. 4, pp. 521-544
- [12] Ng V, Cardie C, “Improving machine learning approaches to coreference resolution”, *In Proceedings of the 40th annual meeting on association for computational linguistics 2002 Jul 6*, pp. 104-111.
- [13] Wiseman S, Rush AM, Shieber SM, “Learning global features for coreference resolution”, *arXiv preprint arXiv:1604.03035*. 2016 Apr 11.
- [14] Lee C, Jung S, Park CE, Anaphora Resolution with Pointer Networks, *Pattern Recognition Letters*, 2017 May 19.
- [15] Hardmeier C, Nakov P, Szymne S, Tiedemann J, Versley Y, Cettolo M, “Pronoun-focused MT and cross-lingual pronoun prediction: Findings of the 2015 DiscoMT shared task on pronoun translation”, *In Second Workshop on Discourse in Machine Translation (DiscoMT), 17 September 2015, Lisbon, Portugal 2015*, pp. 1-16.
- [16] Kilicoglu H, Roseblat G, Fiszman M, Rindfleisch TC, “Sortal anaphora resolution to enhance relation extraction from biomedical literature”, *BMC bioinformatics*, 2016 Apr 14, vol. 17, no. 1, pp. 163.
- [17] Bell D, Hahn-Powell G, Valenzuela-Escárcega MA, Surdeanu M, “Sieve-based Coreference Resolution in the Biomedical Domain”, *arXiv preprint arXiv:1603.03758*, 2016 Mar 11.
- [18] Choi MJ, Coreference resolution for biomedical pathway data (Doctoral dissertation) 2017.
- [19] Loáiciga S, Wehrli E, “Rule-based pronominal anaphora treatment for machine translation”, *In Proceedings of the Second Workshop on Discourse in Machine Translation 2015 Sep 17*, pp. 86-93, Association for Computational Linguistics Lisbon, Portugal.
- [20] van der Aa H, Leopold H, Reijers HA, “Detecting inconsistencies between process models and textual descriptions”, *In International Conference on Business Process Management 2015 Aug 31*, pp. 90-105, Springer, Cham.
- [21] Schumacher P, Minor M, Schulte-Zurhausen E, “Extracting and enriching workflows from text”, *In Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on 2013 Aug 14*, pp. 285-292, IEEE.
- [22] Fajar, Ahmad Nurul, and Imam Marzuki Shofi, “Goal Model to Business Process Model: A Methodology for Enterprise Government Tourism System Development”, *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 6, 2016.
- [23] Herbst J, Karagiannis D, “An inductive approach to the acquisition and adaptation of workflow models”, *In Proceedings of the IJCAI 1999 Aug*, vol. 99, pp. 52-57.
- [24] Blumberg R, Atre S, “The problem with unstructured data”, *Dm Review*, 2003 Feb, vol. 13, 42-49, 62.
- [25] Laport V, Borges MR, Braganholo V, “Athena: A collaborative approach to requirements elicitation”, *Computers in Industry*, 2009 Aug 31, vol. 60, no. 6, pp. 367-380.
- [26] Riefer M, Ternis SF, Thaler T, “Mining process models from natural language text: A state-of-the-art analysis”, *Multikonferenz Wirtschaftsinformatik (MKWI-16)*, March, pp. 9-11
- [27] Pittke F, Leopold H, Mendling J, “When language meets language: Anti patterns resulting from mixing natural and modeling language”, *In International Conference on Business Process Management 2014 Sep 7*, pp. 118-129, Springer, Cham.
- [28] Versley Y, Ponzetto SP, Poesio M, Eidelman V, Jern A, Smith J, Yang X, Moschitti A, “BART: A modular toolkit for coreference resolution”, *In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Demo Session 2008 Jun 16*, pp. 9-12, Association for Computational Linguistics.
- [29] Stoyanov V, Cardie C, Gilbert N, Riloff E, Buttler D, Hysom D, “Reconcile: A coreference resolution research platform”, 2010 Apr 13.
- [30] Toutanova K, Klein D, Manning CD, Singer Y, “Feature-rich part-of-speech tagging with a cyclic dependency network”, *In Proceedings of the 2003 Conference of the North American Chapter of the Association for*

- Computational Linguistics on Human Language Technology-Volume 1 2003 May 27*, pp. 173-180, Association for Computational Linguistics.
- [31] Chen D, Manning C, “A fast and accurate dependency parser using neural networks”, *InProceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) 2014*, pp. 740-750.
- [32] Wu Z, Palmer M, “Verbs semantics and lexical selection”, *InProceedings of the 32nd annual meeting on Association for Computational Linguistics 1994 Jun 27*, pp. 133-138, Association for Computational Linguistics.