

Reducing Event Variability in Logs by Clustering of Word Embeddings

David Sánchez-Charles¹, Josep Carmona², Victor Muntés-Mulero¹, and Marc Solé¹

¹ CA Strategic Research, CA Technologies, Barcelona, Spain,
{David.Sanchez,Victor.Muntes,Marc.SoleSimo}@ca.com

² Universitat Politècnica de Catalunya, Barcelona, Spain.
jcarmona@cs.upc.edu

Abstract. Several business-to-business and business-to-consumer services are provided as a human-to-human conversation in which the provider representative guides the conversation towards its resolution based on her experience, following internal guidelines. Several attempts to automatize these services are becoming popular, but they are currently limited to procedures and objectives set during design step. Process discovery techniques could provide the necessary mechanisms to monitor event logs derived from textual conversations and expand the capabilities of conversational bots. Still, variability of textual messages hinders the utility of process discovery techniques by producing non-understandable unstructured process models. In this paper, we propose the usage of word embedding for combining events that have a semantically similar name.

Key words: Unstructured processes, process discovery, machine learning, word embedding

1 Introduction

Recent trends in Natural Language Processing and Machine Learning allow machines to understand and answer simple queries in the form of free text. Lots of textual data is still generated for describing actions performed during the execution of procedures or services in which the human interaction is a key component. For instance, software development teams textually describe changes on source code, customer support channels record conversations with customer and the actions performed by the support engineers. Although it is well known in the industry that improving the efficiency of such services and procedures lead to more efficient businesses¹, the Business Process Management arena is behind on applying the most recent developments on Natural Language Processing and Machine Learning.

Process Modelling and Discovery have the potential for advancing towards the creation of models of human-to-human, or human-to-computer, interactions,

¹ For instance, a faster customer support channel leads to lower customer churn rates. <https://www.salesforce.com/blog/2017/03/effective-strategies-to-reduce-customer-churn.html>

in which events would represent the invent of a human action or message. Nevertheless, one of the most frequent assumptions in the literature of business process modelling is that events are well defined (i.e. a set of activities fixed during design time). This assumption is no longer applicable in this context, as events are manually defined by humans and, hence, high variability on the event space is expected.

In this paper, we investigate the problem of event name variability for process discovery and propose an approach to resolve this problem through event log pre-processing. In particular, we introduce an approach for clustering event names based on novel similarity metrics between textual data [14] and, afterwards, create a new refined log by projecting events to the discovered clusters. In Section 2, we describe the problem and a general overview of a solution. Then, in Section 3, we explain the details of our solution which is later validated during Section 4. Related work and a discussion on the work presented are provided in Sections 5 and 6, respectively.

2 Log abstraction via Event Variability Reduction

Due to the inherent freedom of language and communication, a textual description of a human activity may never be reused for two executions. Analysis of human-described events is hindered by such variability. In this section, we define an event log abstraction consisting as an event projection that generalizes sets of messages. Its main objective is to reduce the number of distinct events and increase the ratio of shared events among conversations. Later in section 3, we explain how this generalization can consider the semantic similitude between event names.

Support Engineer: How can we help you?

Customer: My installation of Product A is not working properly, the system prompts me an error code TEST001.

Support Engineer: Please, try the fix provided in KB00001.

Fig. 1. Example of a conversation between a Customer and a Support Engineer. Each interaction in the conversation is considered as a text-based event with an event name consisting of the factual textual message. In this example, the trace consists of three events.

We define² the alphabet MSG of all the possible messages that may be interchanged in a conversation. We will assume that this alphabet is formed by words, sentences and paragraphs, albeit other non-textual messages may be interchanged. A **text-based event** is any instance of an element of the alphabet

² For the sake of simplicity, the definitions and examples of the paper are tailored to the context of conversations between humans and, possibly, computers. In spite of this, the theory of the paper can be applied to general event logs as defined in [18].

MSG, and a **conversation** is a trace of text-based events. I.e. a non-empty sequence of instances of elements in *MSG*. Finally, an **event log** is a collection of conversations of a conversational-based human-to-human service . Figure 1 depicts an example of a typical conversation considered in this paper.

Definition 1. Given an alphabet *A*, a log *L* with the set of traces defined in *A* and a surjective mapping α between the alphabet *A* and an alphabet *B*, the **Event Variability Reduction (EVR)** is the projected event log

$$L' = \{\langle \alpha(e_0), \alpha(e_1), \dots, \alpha(e_n) \rangle \mid \langle e_0, e_1, \dots, e_n \rangle \in L\}$$

Generally speaking, the EVR replaces all repetitions of an event *e* by its image $\alpha(e)$. Due to the surjection of the mapping α , the size of *B* is smaller than the original alphabet *A* and, hence, we are performing a reduction of the event space. Although performing such abstraction reduces the information contained in the event log, it enables practitioners to compare different traces. In fact, the major benefit of the EVR technique is when coupled with other techniques. In Section 4, we will evaluate the combination of EVR with process discovery and sequence classification techniques.

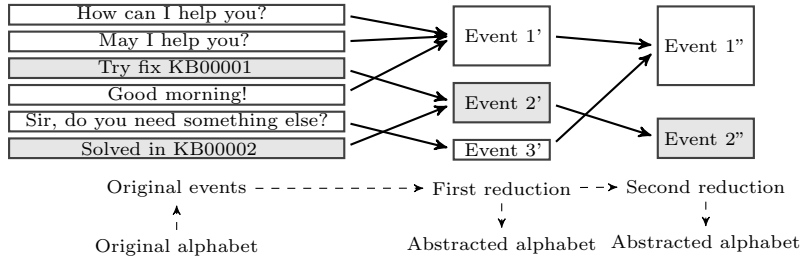


Fig. 2. Graphical representation of two executions of an EVR method over 6 fictional textual messages. The color of the box depicts the final abstract event.

Figure 2 depicts a graphical example of the application of EVR techniques to the alphabet of an event log *L*, which is comprised by 6 distinct messages in a conversation. A first run of the EVR technique reduces the number of distinct messages to 3, and generated an abstracted alphabet. The resulting event log after the first EVR contains the projected events as specified by the arrows. Notice that the two first events in the abstracted alphabet are now referring to the same abstract *Event 1'* instead of *How can I help you?* and *May I help you?*. A second run of the EVR method simplifies even more the event space to only 2 distinct events.

3 Approach

The utility of the EVR is based on the quality of the event set mapping that we consider. In general, it is expected that if two events e_1 and e_2 are projected

into the same event e , then both events have a property in common that is not necessarily shared with events not projected to e . Assuming that the name of the events is a good representative of the real action performed by a human, we propose to group together those events that have similar event names. We will use a novel technique, explained in Section 3.1, for measuring the similarity of two event names. Such similarity compares the semantics of words and sentences instead of the exact repetition of words as in traditional bag-of-words techniques.

Figure 3 summarizes our methodology. First, we retrieve all event names from the log and we compute the embeddings of all the words contained in the event name, as specified in Section 3.1. These word embeddings allow us to compute a word similarity, that is later averaged for measuring a similarity metric between event names such that text-based events with similar semantics are very similar according to this metric. Finally, we consider a clustering technique for discovering group of events and we use this as the event set partition of the embedding-based EVR.

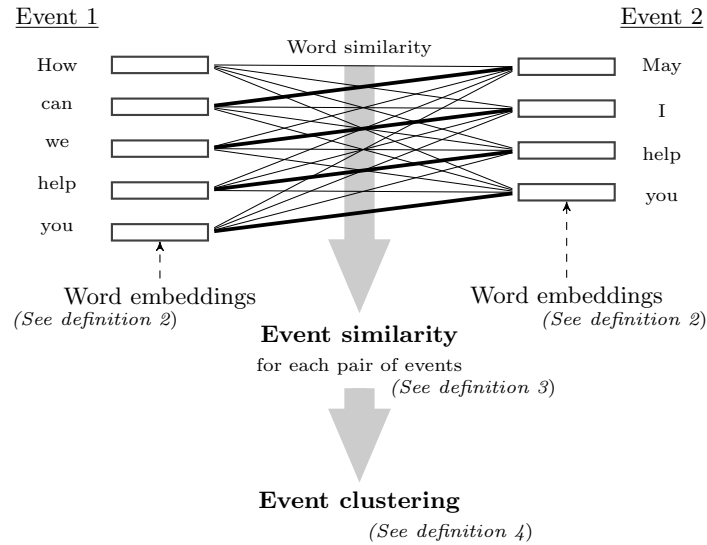


Fig. 3. Overview of the Event Variability Reduction based on word embeddings. A clustering technique utilizes an embedding-based text similarity for creating groups of events.

3.1 Word Embeddings

An embedding is a function that generates representations of objects difficult to analyze into a well-known space, such as a vector space, allowing further analysis. In this paper, we focus on word embeddings.

Definition 2 (Bengio et.al. [5]). *A word feature vector or **word embedding** is a function that converts words into points in a vector space. Word embeddings are usually injective functions (i.e. two words do not share the same word embedding), and highlight not-so-evident features of words. Hence, one usually says that word embeddings are an alternative **representation** of words.*

Word2Vec [14] is a word embedding in which words are mapped into a fixed-length vector space, such that the cosine similarity of the embedding of two words is a good estimator of its semantic similarity. The major benefit of using Word2Vec is that the training method does not need to manually build or validate complex taxonomies, but it learns by extracting the *meaning* of a word by considering its adjacent words in a set of sentences. Typically, a large textual corpora such as Wikipedia is considered, but it could also leverage information from knowledge-specific documentation. Moreover, accuracy with respect to unsupervised count-based techniques [4] positions Word2Vec as the perfect candidate for measuring similarity of textual data.

Authors in [11] extended the results obtained by the Word2Vec technique in order to compute a similarity between short messages. Their approach measures the pairwise similarity between words of the two sentences, averaging by the inverse document frequency³ of words. Although it is out of the scope of this paper, other embedded-based similarity metrics could also be considered [13].

Definition 3 (Tom Kenter and Maarten de Rijke [11]). *Given two event names E_1 and E_2 (with E_1 shorter than E_2), its embedded-based similarity is*

$$Sim(E_1, E_2) = \sum_{w_1 \in E_1} idf(w_1) \cdot \frac{(c + 1) \cdot \max_{w_2 \in E_2} Sim(w_1, w_2)}{\max_{w_2 \in E_2} Sim(w_1, w_2) + c \cdot \left(1 + b - b \cdot \frac{|E_2|}{\text{average length}}\right)},$$

where w_i is a word of the sentence E_i , $Sim(w_1, w_2)$ is the cosine similarity between the Word2Vec embeddings of w_1 and w_2 and b, c are two regularization constants⁴.

3.2 Event Rediscovery via Document Embedding Clustering

In the previous subsection, we have defined a similarity between event names based on a word embedding known as Word2Vec. We propose to use a clustering technique based on this similarity metric in order to retrieve groups of similar event names to discover a set of abstract activities that will be consumed by an EVR.

Definition 4. *Given a log L , with a set of distinct events E , and a partition of the event set $\{E_i\}_{i \in I}$ ⁵ obtained by running a clustering technique on E with*

³ We follow the classical definition $idf(w) = \log \frac{\text{Number of documents}}{\text{Occurrences of } w}$.

⁴ During the evaluation of this approach, we set c to 1.2 and b to 0.75 as proposed by [11]

⁵ i.e. a finite collection of sets $\{E_i\}_{i \in I}$ such that $\cup_{i \in I} E_i = E$ and $E_i \cap E_j = \emptyset$ for any $i \neq j$

the embedding-based similarity metric, we define an **Embedding-based Event Variability Reduction** as the EVR defined over the mapping α such that $\alpha(e) = i \in I$ such that $e \in E_i$.

After applying an embedding-based EVR, the newly discovered event log has reduced the variability of event names by discarding the wording used and, instead, focusing on the semantic of the words. Depending on the clustering technique used and parameters, one could obtain event logs with different levels of granularity. Notice that the event identifier of the newly discovered event log is a set of events in the original event log, the end-user may need to check the set of events to understand the abstracted event log.

4 Evaluation

To evaluate the approach presented in this paper, we chose a dataset in which *event names* hold information about an unknown activity, there are some guidelines on how those *events* should be named and positioned in the trace (i.e. the system process) and results can be easily interpreted. With such dataset, we will perform a preliminary process analysis that would have been impossible without the use of the EVR. Afterwards, we apply the techniques to an industrial dataset comprising textual conversations between technical support engineers and customers. Prior to the technique described in the paper, the lack of structure in textual messages did not allow for a mechanism to monitor the evolution of conversations.

4.1 Structure of Documents in Wikipedia

Mass collaboration projects usually rely on guidelines to palliate the variability of human outcomes, and sub-communities are created to ensure better coherence. Wikipedia is a great example of such a complex collaboration project, with over 200 guidelines⁶, ranging from behavioral recommendations on a discussion to naming rules. We hypothesize that the Table of Contents of Wikipedia articles follows some of these guidelines. We will evaluate how the embedding-based EVR helps process discovery techniques in discovering such guideline.

Discovery of the Structural Process Model We retrieved over 800 articles from Wikipedia⁷, selected from the list of featured articles⁸ of the *Media, Literature and Theater, Music biographies, Media biographies, History biographies* and *Video gaming* categories. From the list of articles we extracted the structure of the document, i.e. sections and subsections of the text.

Directly applying process discovery techniques over such event logs generates the *flower model*⁹ in all cases, primarily due to the high ratio of distinct

⁶ https://en.wikipedia.org/wiki/Wikipedia:List_of_guidelines

⁷ 8th August 2016. The dataset is publicly available on data.4tu.nl [16].

⁸ https://en.wikipedia.org/wiki/Wikipedia:Featured_articles

⁹ The flower model is a model that allows any possible behavior.

events over events seen in the log. In fact, several events are only seen once and, hence, patterns in the structure of the document are difficult to find. Besides, comparison of traces between categories is almost impossible. To overcome this challenge, we applied the embedded-based EVR technique for discovering a set of 50 abstract events shared among all the Wikipedia articles. After discovering such abstract events, we see a complete different picture allowing us to further analyze this dataset.

Cluster 1	Cluster 2	Cluster 3
in culture philosophy re-discovery mythology historiography	travels first voyages privateering expedition journey trans-pacific voyage	aftermath of centralia history of the manuscripts with the five the poems in the media
Cluster 4	Cluster 5	Cluster 6
writing and publishing writing career writing style writing history writing	return to united states return to france return to missouri return to canada return to japan	lineups collaborations recording discography collaborators

Table 1. 5 randomly chosen section titles from the first 6 out of 50 discovered clusters from the Wikipedia dataset.

Table 1 shows six activities discovered by applying the embedded-based EVR. One may check that Cluster 4 trivially refers to sections involving *writing*, and Cluster 5 combines sections containing *return to*. Nevertheless, other combinations are less trivial such as the sections included in Cluster 2. Unfortunately, some groups are not as accurate as the aforementioned. For instance, the first cluster seems to contain topics related to *philosophy*, *religion* and *history*. The three topics are certainly related, but a better granularity on such cluster might be necessary.

The embedded-based EVR replaces all the listed Wikipedia titles in Table 1 with the assigned Cluster number. Therefore, one may take the set of titles as the new *event name*. This may hinder the understandability of the discovered abstract activities, as the practitioner needs to take a look into the clustered items, as we have done in the above paragraph, to have an understanding of their relation.

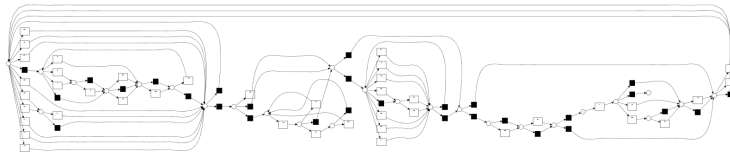


Fig. 4. Petri net discovered from articles in the Music biography category.

Continuing the log analysis, we run a process discovery method on each of the logs. In particular, we run the Inductive Miner¹⁰. An example of a process model discovered after applying the EVR is depicted in Figure 4. Most of them have a small subprocess with a flower-like behavior, and an in-depth analysis of the traces and abstract events highlighted several section and subsections with similar names that may happen in any ordering. Nevertheless, in general, a more detailed pattern has been found in this dataset thanks to the event abstraction.

		History	Media	Music	Literature	Video	
		biography	biography	biography		games	Media
History	F	0.94	0.79	0.72	0.79	0.65	0.68
biography	P	0.37	0.37	0.37	0.39	0.41	0.43
Media	F	0.73	0.91	0.77	0.69	0.59	0.61
biography	P	0.47	0.42	0.46	0.49	0.50	0.52
Music	F	0.88	0.92	0.97	0.86	0.69	0.72
biography	P	0.35	0.34	0.34	0.36	0.38	0.39
	F	0.71	0.66	0.66	0.81	0.75	0.80
Literature	P	0.41	0.39	0.45	0.48	0.46	0.48
Video	F	0.44	0.49	0.55	0.64	0.87	0.73
games	P	0.66	0.64	0.65	0.64	0.55	0.65
	F	0.74	0.75	0.75	0.83	0.97	0.93
Media	P	0.43	0.42	0.44	0.42	0.43	0.46

Table 2. Table summarizing quality of the discovered process models. For each row, fitness and precision of a process model is measured with respect to all the logs.

Comparison of the Structure on Wikipedia Articles The process model depicted in Figure 4 is a first approach to find the underlying guideline for writing articles in their respective categories. Table 2 depicts alignment-based fitness [2] and precision [1] of the discovered process models with respect to all the abstract event logs, and serves as a mechanism to compare the underlying guidelines between the categories. One may notice that the three *biography* process models have high fitness with all the biography logs, but it is significantly lower with respect to *Video gaming* and *Media* categories. The contrary also holds, as fitness of the *Video gaming* and *Media* process models is significantly lower with respect to the three biography logs. On the other hand, the *Literature* category fits fairly well in both groups.

These results were expected, as all *biography* articles should have a similar structure (although talking about different types of artists or personalities) and *videogames* are nowadays produced as popular films and series (as they appear in the *media* category). On the contrary, *literature* articles usually talk about the authors and historical context of the book, and also about its plot (which is very common in *videogames* and *media* articles).

¹⁰ We run the infrequent version of the Inductive Miner, with default parameters, on ProM 6.5.1.

4.2 Application of the Event Variability Reduction to Trace Monitoring of Human-driven Processes

Some textual documents such as Tickets in Support systems, or live support chats, evolve over time. For example, tickets consist of a sequence of messages exchanged between a customer and one or more support engineers. The first messages usually provide a first description of the problem. Nevertheless, the content may evolve throughout the chain of messages and derive to other topic as the root-cause of the issue is being discovered. When the conversation between the customer and the support team ends, the ticket is usually enriched with extra information about the conversation outcomes such as the product causing the issue, type of fix needed, solution proposed, time to complete the issue, or satisfaction of the client with the support team. If this final information is known during the conversation, a support engineer would be able to better guide the conversation.

It is very important for the industry to predict the level of satisfaction of a customer with respect to the service offered. Customer escalation is the formal mechanism that customers have to warn support engineers that the resolution of an issue is not as fast and smooth as they expected. In fact, the number of escalations is used as a Key Performance Indicator (KPI) for measuring the quality of support teams, and it is clearly an indicator of customer dissatisfaction and churning.

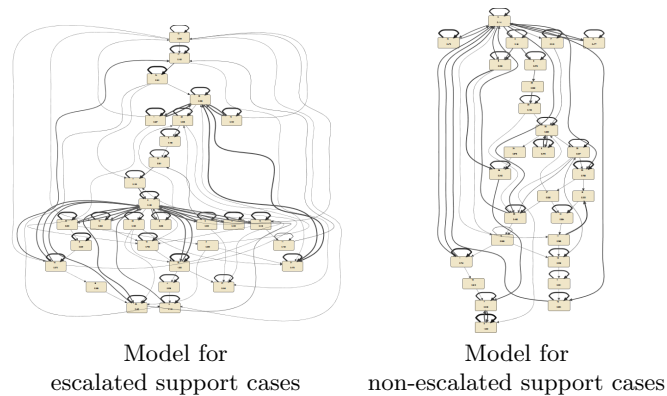


Fig. 5. Fuzzy Nets obtained from the support cases after applying the embedded-based EVR. Escalated cases are more unstructured than non-escalated cases, indicating that either support engineers need a broader exploration of the issue or that customers felt that the case was not properly handled.

We applied the EVR technique in an industrial dataset provided by CA Technologies. This dataset contains the messages interchanged between support engineers and customers during 2015, as well as all customer escalations during the same period. We applied the EVR technique for discovering models specific

to escalated support cases and non-escalated cases, with the objective of building a predictor for future cases. Figure 5 depicts the fuzzy models [19] of both categories. Notice the structural difference between the two process models. The escalated process model is more unstructured than the non-escalated process models. This might indicate that the support engineers need a broader exploration phase of such cases.

We trained a pair of Hidden Markov Models [7] for building a predictor capable of classifying non-complete traces. At the initialization step, we used the process models in the two figures as the structure of the hidden states. Then, probabilities were tuned during the training of the model for maximizing the accuracy of the classifier. Despite the structural difference provided by the fuzzy models and the existence of differential, albeit not-so-frequent, small sequential patterns, roughly 10% of accuracy was achieved when detecting escalated cases¹¹. The high imbalance between the escalated and non-escalated datasets may have caused the low ratio of detected escalations. Besides, it also indicates that escalations are not primarily caused by a global property of the conversation, but other external features must be taken into account. Nevertheless, it has been acknowledged by the company as a useful tool as it reduces the number of cases that need to be closely monitored for having an impact on customer satisfaction.

5 Related work

Different approaches exist in the literature for the problem of discovery and management of process models with a large set of supported activities. For instance, the Fuzzy Miner [19] allows practitioners to choose a level of abstraction for the discovered process model, and the algorithm automatically merges different events into a single, and more abstract, cluster of events. Nevertheless, these approaches are either based on the *directly-follows* relation [19] [8], a temporal correlation [9] or satisfying a particular known pattern [17] (or initially unknown patterns [6]). Our approach is not based on the fact that events may have a sequential or temporal relation between them, but that the event name similarity indicates how similar are two events.

The art of modelling human conversations, known as Speech Acts or Dialog Acts, is a well known challenge in Computer Science [15]. Nevertheless, to the best of our knowledge, current applications of Speech Acts follow a top-down approach in which textual messages are classified based on a list of known Intents¹². As an example, Authors in [10] combine Intent detection for abstracting Events and then measures the likelihood of changing from one Intent to another.

Bag-of-words techniques, i.e. using the frequency of each word in a document, have been largely used for comparing two texts and discover a list of

¹¹ Results are consistent with respect to a 20%-out cross-validation.

¹² ISO 224617-2 defines 57 generic communicative functions, that one may enrich or refine depending with domain knowledge.

topics in a set of documents. Process Matching [12] has considered these techniques for matching activities of different known processes, and [3] consumes activity names, and their descriptions, to map events to activities of a known process model. None of them considers event name abstraction for the discovery of the process model. Besides, the approach presented in this paper enables the comparison of word semantics instead of considering exact word matches.

6 Discussion

In this paper, we have developed a method for reducing variability of event names by grouping them according to their similarity. Recent developments on Natural Language Processing allowed us to compute this similarity based on semantic information, instead of traditional bag-of-words techniques or creating a complex ontology.

We applied this technique on a dataset compromising the structure of articles in Wikipedia. Initially, it was impossible to find any common structure because section names were almost never repeated in the dataset. Nevertheless, after applying EVR, common process discovery technique already discovered some patterns on the data and enable us to compare two different articles. Although this use case is very simplistic, the results validate the methodology and motivates further research in this direction. We have also used this technique for analyzing the structure of support cases, and we arrived to the conclusion that there is a weak relation between customer satisfaction and the content of the sequence of messages interchanged between support engineers and customers.

This paper focus on the event name similarity of two events, but it does not consider the frequency of the events nor its role in the trace. Further research should consider how this information can be leveraged to discover better abstract events.

Acknowledgements. This work is funded by Secretaria de Universitats i Recerca of Generalitat de Catalunya, under the Industrial Doctorate Program 2013DI062, and the Spanish Ministry for Economy and Competitiveness, the European Union (FEDER funds) under grant COMMAS (ref. TIN2013-46181-C2-1-R).

References

1. A. Adriansyah, J. Munoz-Gama, J. Carmona, B. F. Dongen, and W. M. Aalst. Measuring precision of modeled behavior. *Inf. Syst. E-bus. Manag.*, 13(1):37–67, February 2015.
2. A. Adriansyah, B. F. van Dongen, and W. M. P. van der Aalst. Conformance checking using cost-based fitness analysis. In *Proceedings of the 2011 IEEE 15th International Enterprise Distributed Object Computing Conference*, EDOC '11, pages 55–64, Washington, DC, USA, 2011. IEEE Computer Society.

3. Thomas Baier, Jan Mendling, and Mathias Weske. Bridging abstraction layers in process mining. *Information Systems*, 46:123–139, 2014.
4. Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of Association for Computational Linguistics (ACL)*, volume 1, 2014.
5. Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.
6. RP Jagadeesh Chandra Bose and Wil MP van der Aalst. Abstractions in process mining: A taxonomy of patterns. In *International Conference on Business Process Management*, pages 159–175. Springer, 2009.
7. Gil Aires Da Silva and Diogo R Ferreira. Applying hidden markov models to process mining. *Sistemas e Tecnologias de Informação. AISTI/FEUP/UPF*, 2009.
8. Christian W Günther, Anne Rozinat, and Wil MP Van Der Aalst. Activity mining by global trace segmentation. In *International Conference on Business Process Management*, pages 128–139. Springer, 2009.
9. Christian W Günther and Wil MP van der Aalst. *Mining activity clusters from low-level event logs*. Beta, Research School for Operations Management and Logistics, 2006.
10. Zhiyang He, Xien Liu, Ping Lv, and Ji Wu. Hidden softmax sequence model for dialogue structure analysis. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.
11. Tom Kenter and Maarten de Rijke. Short text similarity with word embeddings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM ’15*, pages 1411–1420, New York, NY, USA, 2015. ACM.
12. Christopher Klinkmüller, Ingo Weber, Jan Mendling, Henrik Leopold, and André Ludwig. Increasing recall of process model matching by improved activity label matching. In *Business Process Management*, pages 211–218. Springer, 2013.
13. Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1188–1196, 2014.
14. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.
15. RA Morelli, JD Bronzino, and JW Goethe. A computational speech-act model of human-computer conversations. In *Bioengineering Conference, 1991., Proceedings of the 1991 IEEE Seventeenth Annual Northeast*, pages 263–264. IEEE, 1991.
16. D. Sanchez-Charles. Title and subtitles of wikipedia articles, 2017. doi:10.4121/uuid:61fb9665-40ab-4b70-8214-767c521cc950.
17. Niek Tax, Natalia Sidorova, Reinder Haakma, and Wil M. P. van der Aalst. Event abstraction for process mining using supervised learning techniques. *CoRR*, abs/1606.07283, 2016.
18. Wil M. P. van der Aalst. *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer, 2011.
19. Wil M. P. van der Aalst and Christian W. Günther. Finding structure in unstructured processes: The case for process mining. In *ACSD*, pages 3–12. IEEE Computer Society, 2007.