Clustering Amendments with Semantic Embeddings

Alessandro Sajeva[1,*], Stefano Iannucci[1], Carlo Marchetti[2], Paolo Merialdo[1] and Riccardo Torlone[1]

[1]*Università degli Studi Roma Tre*
[2]*Senato della Repubblica Italiana*

Abstract
The Italian Senate faces the problem of clustering amendments to optimize the scheduling of parliamentary sessions. Currently, this task is carried out by Similis, an application that tackles this problem by using a traditional term-frequency technique, which leads to clustering based on wording rather than semantics. Recent advances in natural language processing have led Italian institutions to investigate the adoption of pre-trained language models (PTLMs) for text analysis. Along this line, in this paper, we propose CLAMSE, an alternative system to Similis that uses Sentence-BERT pre-trained models to generate embeddings and then groups similar amendments through hierarchical agglomerative clustering. Our preliminary evaluation shows that CLAMSE achieves comparable performance to Similis using embeddings generated by pre-trained models without fine-tuning, paving the way for applying a clustering method with advanced contextual understanding. This study contributes to enhancing the effectiveness of institutional decision-making processes through the adoption of PTLMs.

Keywords
language models, embeddings, clustering, public affairs

1. Introduction

Pre-Trained Language Models (PTLMs) are emerging as valuable allies in addressing various problems across diverse domains and represent a great opportunity for enhancing parliamentary efficiency and effectiveness. In this context, and within the framework of a collaboration between Roma Tre University and the Italian Senate, an interest in systems based on PTLMs to support parliamentary activities has taken shape. This paper investigates the problem of clustering similar amendments.

Amendments represent proposed changes to legislative texts and are a fundamental element of the legislative process. They may vary widely in terms of wording but often share similar intentions and objectives. Similar amendment proposals should be discussed simultaneously, if possible. Therefore, clustering amendments according to their similarity is an essential activity to facilitate the work of officials and effectively organize voting sessions, while ensuring the coherence and completeness of legislative proposals. Indeed, amendments that differ by only

SEBD 2024: 32nd Symposium on Advanced Database Systems, June 23-26, 2024, Villasimius, Sardinia, Italy
*Corresponding author.
alessandro.sajeva@uniroma3.it (A. Sajeva); stefano.iannucci@uniroma3.it (S. Iannucci); carlo.marchetti@senato.it (C. Marchetti); paolo.merialdo@uniroma3.it (P. Merialdo); riccardo.torlone@uniroma3.it (R. Torlone)
0009-0000-9419-3015 (A. Sajeva); 0000-0001-7485-9772 (S. Iannucci); 0000-0002-3852-8092 (P. Merialdo); 0000-0003-1484-3693 (R. Torlone)

a few words are usually proposed in large numbers by parliamentary groups that want to filibuster, with the aim of slowing down the legislative process. Combining debates on similar amendment proposals therefore allows for the greatest possible efficiency in terms of time.

In this paper, we aim to explore the potential of PTLMs in such a crucial context. The Senate already has a tool to support this activity, called Similis, which adopts a traditional term-frequency technique. Although Similis is effective in grouping short amendments sharing many tokens, it loses effectiveness with longer amendments that adopt different lexicons yet preserve the same semantics.

To overcome this issue we have investigated an alternative solution that leverages a PTLM. Our approach has been implemented in CLAMSE (Clustering Amendments with Semantic Embeddings), an alternative system to Similis, which relies on Sentence-BERT [1] PTLMs for converting amendments into embeddings, and then groups similar embeddings via a hierarchical agglomerative clustering (HAC) technique.

The preliminary experiments we conducted yielded promising results showing the effectiveness of adopting solutions based on PTLMs in demanding processes of public administration, with the advantage of simplifying the development process by eliminating the need for building from-scratch implementations or task-specific models.

The rest of the paper is organized as follows. In Section 2 we discuss related works and the pre-existent approach to amendments clustering. In Section 3 we illustrate our PTLM solution and in Section 4 we report on its evaluation. Finally, in Section 5 we draw some conclusions.

2. Related work and earlier solution

In recent years, there has been a growing emphasis on leveraging advanced text processing techniques to enhance institutional work globally. In the context of a collaboration between Roma Tre University and the Senate of the Italian Republic, and more generally in the application of artificial intelligence systems to support legislative activities, several important studies have been carried out. A machine learning system for the classification of documents has been developed [2], and another important contribution concerned the implementation of a system that exploits Sentence-BERT [1] for the alignment of stenographic reports with audio recordings of parliamentary sessions [3], representing steps forward in the digital transformation of legislative practices.

A pivotal aspect of this evolution lies in the transition from traditional vectorization methods towards more semantic-based approaches. In the domain of text clustering, Term Frequency-Inverse Document Frequency (TF-IDF) stands out as one of the most commonly employed methods for representing textual data. However, TF-IDF fails to incorporate the positional and contextual aspects of words within sentences. A study conducted simulations to demonstrate that BERT consistently outperforms the TF-IDF method [4]. Furthermore, recent studies have employed Sentence-BERT as a vectorization technique before clustering analysis [5]. Comparisons have been made between Sentence-BERT and traditional methods. Sentence-BERT emerged as particularly adept at understanding topics within clustering algorithms [6]. Other research works also have showcased the effectiveness of semantic embeddings derived from foundation models in clustering endeavors in the realm of dataset deduplication [7].

This transition towards semantic representations marks a paradigm shift from shallow representations incorporating syntactic meaning to deep context understanding, providing a more sophisticated text comprehension.

Similis. The IT department of the Italian Republic Senate has developed a solution for clustering similar amendments, called Similis, in close collaboration with the Institute of Legal Informatics and Judicial Systems (CNR-IGSG) [8]. The similarity sought by Similis focuses on the wording of sentences, thus favoring texts with high syntactic rather than semantic coherence. The algorithm does not rely on a priori information about amendments, and groups amendments by means of HAC with complete linkage [9]. The content of each amendment is represented by a vector which is built according to a bag-of-word model with term-frequency (TF) with Euclidean normalization [10], after word stemming and stop-words removal. Amendment similarity is measured using a traditional cosine similarity metric. The thresholds for the cosine similarity and dendrogram cut-off have been identified empirically and are set to 0.8 and 0.2, respectively.

3. CLAMSE

Our semantic embeddings-based solution consists of a pipeline, which involves three phases: (*i*) preprocessing, (*ii*) encoding, and (*iii*) clustering.

The system takes a set of amendments that refer to a single Senate act as input. The preprocessing phase aims at cleaning the text of each amendment from special characters, thereby increasing the quality of embeddings and, consequently, improving clustering performance. The pre-processed text is then sent to the encoding block, which computes an embedding for each amendment. It is worth saying that Sentence-BERT models are implemented in a Python framework known as SentenceTransformers. This provides several PTLMs, each with specific characteristics. The encodings produced by each model serve as input for the clustering phase that implements a traditional HAC. The final solution for the problem is determined by selecting the most optimal clustering among those generated by applying HAC to each corpus of embeddings obtained in the encoding phase.

In the following, we provide a more detailed description of the three phases.

3.1. Preprocessing

The input dataset, which is structured as a JSON Line (JSONL) file, is transformed into the standard JSON format for readability. The JSON file reports a record containing the text and the amendment number for each amendment. Also, each record is associated with a cluster identifier, which represents the ground truth for a clustering assignment.[1]

The text of the amendments includes numerous tags and annotations designed for display on web browsers but lacking any semantic significance. Therefore, it was decided to remove them to enhance clustering performance. In particular, we remove all the occurrences of the HTML

[1]In the original JSON file, these elements correspond to the 'num_em', 'text_emend', and 'id_cluster' attributes

Table 1
Sentence-BERT PTLMs used in our experimental activity: acronym, extended name, size (in MB), dimension of the embedding, and max sequence length

model	extended name	size	dimension	max sequence length
AMB1	all-mpnet-base-v1	420	768	512
GTXL	gtr-t5-xl	2370	768	512
MQMBC1	multi-qa-mpnet-base-cos-v1	420	768	512
MQMBD1	multi-qa-mpnet-base-dot-v1	420	768	512
PMB2	paraphrase-mpnet-base-v2	420	768	512

macro for the blank space (”&nbsp”), and we replace all the occurrences of HTML tags with whitespace characters.

3.2. Amendment encoding with Sentence-BERT

The cleaned corpus of amendments is passed to the encoding module. Here, a range of Sentence-BERT PTLMs is loaded, and each model generates an embedding for the text of each amendment. Not all of the pre-trained models produce normalized embeddings. However, since normalizing embeddings leads to improved performance, a normalization step is added by dividing each component of the vector by its norm and ensuring that the norm of the resulting vector is equal to 1 (L2 normalization). Table 1 reports the PTLMs that we have considered in our experimental activities. Since the dataset used for the experiments contains amendments with an average of 137 tokens, models with a maximum sequence length of less than 512 tokens were excluded from the study. [2]

3.3. Clustering algorithm

In the last phase, a clustering activity is carried out on each corpus of embeddings generated in the previous phase. In particular, we adopt a HAC approach. Roughly speaking, a HAC algorithm operates as follows. Initially, each sample in the input dataset is treated as an individual cluster. Then, the algorithm iterates, merging in each step the most suitable pair of clusters until only one cluster remains. The decision on which pair of clusters to merge is based on the cosine similarity between the embeddings representing the amendments, with complete linkage, i.e., the similarity between clusters equals the similarity between the two most dissimilar samples, one in each cluster.

The cut-off threshold for the dendrogram generated by the HAC algorithm is set to a value that maximizes the quality of the clusters. In CLAMSE, we use the silhouette score to measure clustering quality: it measures how similar an element is to members of the cluster it belongs to, compared to other members of other clusters. It ranges from -1 to $+1$, where a high value indicates that there is cohesion between elements of the same cluster and separation between elements of different clusters.

[2]For a complete list of the models see https://www.sbert.net/docs/pretrained_models.html.

Table 2
ground truth datasets “Act 1248”

Number of amendments	1.261
Average number of tokens per amendment	137.1
Max number of tokens	6,756
Min number of tokens	3
Number of clusters	664
Average number of amendments per cluster	1.899
Size of the largest cluster	27
Size of the smallest cluster	1

Since we have a clustering solution for each PTLM used to generate the embeddings, we choose the clustering with the highest silhouette score as the final solution.

4. Experimental evaluation

The evaluation of CLAMSE is carried out in terms of both its absolute performance and in comparison to the pre-existing Similis solution. In particular, we first analyze the performance of the different PTLMs to identify the best solution that CLAMSE could achieve. Then, we compare CLAMSE to Similis. Finally, we present the results of an experiment that shows how the preprocessing phase can influence the final clustering results.

4.1. Ground truth

We evaluated CLAMSE on a set of amendments, named “Act 1248”. It contains 1261 amendments that have been clustered manually (each amendment, in the original JSONL file, is annotated with a label representing the target cluster). Table 2 reports the main characteristic of this dataset. Note there is a large variation in both the size of amendments occurring in such dataset (ranging from a few to thousands of tokens) and the number of elements occurring in a cluster (ranging from a singleton to tens of amendments). This makes the task of automatic amendments clustering challenging.

4.2. Evaluation metrics

The performance of Similis is reported by means of the Adjusted Rand-Index (ARI) [11, 12, 13] and the Adjusted Mutual Information (AMI) [14, 15], which are standard metrics to evaluate the results of a clustering process.

ARI assesses the agreement between the clusters produced by a clustering algorithm and the ground truth, correcting for chance agreement. AMI is another measure used to evaluate the similarity between two clusterings of data, but it is based on mutual information. Both ARI and AMI are used to measure the similarity between the true labels in the ground truth and the clustering labels produced by CLAMSE, taking into account the possible presence of random agreements. Both metrics range from 0 to 1, where a score of 1 indicates that true labels and clustering labels are identical, and a score of 0 indicates that they are completely different.

ARI is more suitable when the ground truth clustering has large equal-sized clusters. AMI is preferable when the ground truth clustering is unbalanced and there exist small clusters [16].

As above mentioned, silhouette is another metric for evaluating the quality of clustering. However, in our solution, it is used internally to the HAC algorithm to find the optimal number of clusters.

4.3. Comparison with Similis

The performances of CLAMSE compared to Similis are reported in Table 3, which shows that CLAMSE achieves better results both for ARI and AMI.

Table 3 reports also the performance of CLAMSE without the preprocessing phase. It is worth observing how cleaning text contributes to improved performance as the semantics of the amendments can thus be better understood by the model.

Table 3
Comparison CLAMSE with Similis on Act 1248

model	AMI	ARI
CLAMSE	0,73485	0,58938
Similis	0,71476	0,34511
CLAMSE (no preprocessing)	0,54369	0,32997

4.4. Robustness of CLAMSE

As we discussed in Section 3.2, CLAMSE utilizes several PTLMs from the Sentence-BERT library, known as SentenceTransformers, and chooses the solution that produces the best clustering based on the silhouette score. In order to evaluate the robustness of the approach, for each Sentence-BERT PTLM we have computed the best clustering that could be obtained from the HAC in terms of ARI and AMI. Essentially, at every iteration of the HAC algorithm, we compute ARI and AMI, which need the ground truth. The highest AMI and AMI scores represent the best performance that can be achieved by the CLAMSE approach, which chooses the best clustering of each PTLM based on the silhouette scores.

Table 4 shows, sorted by descending AMI, the best performances achievable by CLAMSE with each of the models in Tables 1. Observing the results of this experiment, two important observations occur. First, notice that in many cases, the CLAMSE algorithm achieves the best results it could obtain. This demonstrates the effectiveness of the silhouette score for choosing the best clustering of each model. Secondly, it's important to note that while several models outperform Similis, there are also many models that exhibit inferior performance. This underscores the effectiveness of our approach in model selection.

5. Conclusions

We presented CLAMSE, a system that addresses the problem of amendments clustering. CLAMSE applies a HAC algorithm to the embeddings of the amendments built by several

Table 4
Silhouette, ARI and AMI reached through silhouette optimization on Act 1248, sorted by best AMI

model	silhouette	AMI	best AMI	ARI	best ARI
MQMBC1	0,41720	0,75333	0,75333	0,63057	0,63057
PMB2	0,42515	0,73485	0,74161	0,58938	0,59005
MQMBD1	0,40534	0,70468	0,73968	0,57445	0,58206
AMB1	0,39858	0,73818	0,73865	0,59765	0,61516
GTXL	0,41664	0,67903	0,70897	0,46131	0,50561

Sentence-BERT PTLMs. This is a preliminary study whose main objective was to explore and evaluate the application of embeddings generated by PTLMs, in particular Sentence-BERT, in comparison to the traditional approach based on TF implemented by the existing system Similis. The preliminary results are encouraging and show that CLAMSE, without fine-tuning, achieved interesting performance.

While these results are promising, several limitations should be noted. First, due to the constraints of using models with a maximum input length of 512 tokens, amendments longer than this threshold are truncated, accounting for approximately 3.33% of the dataset. Second, the encoding is based solely on the textual content of the amendments, ignoring the specific articles of the law to which they refer. As a result, amendments with identical text but referencing different articles may be grouped together in cases where they should not be, potentially compromising the accuracy of the clustering results. In addition, the computational efficiency of the method is a concern, as it requires encoding the entire corpus with multiple models and running HAC each time to determine the optimal clustering result.

This evaluation suggests a promising potential for the future development of CLAMSE. In particular, the possibility of training specific models for amendment clustering could be explored, taking advantage of the pre-trained models that already show good efficiency in terms of semantic embedding. This approach may represent an interesting research direction to further optimize the performance of the system and refine its adaptability to new datasets of amendments to be clustered.

It is worth observing that Similis can leverage Linkoln [17], a system for the automatic extraction of legislative and jurisprudential references from texts in the Italian language. In the future, we plan to study the opportunity of enhancing CLAMSE with the text annotation provided by Linkoln. There exists a pronounced enthusiasm for large language models (LLMs) over traditional PTLMs. Numerous governments are actively experimenting with LLMs for classification tasks and question-answering [18, 19]. Our future endeavors will focus on harnessing the power of LLMs to enhance the performance of CLAMSE.

References

[1] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. arXiv:1908.10084.

[2] A. D. Angelis, V. D. Cicco, G. Lalle, C. Marchetti, P. Merialdo, Multi-label classification

of bills from the italian senate, in: AIxPA@AI*IA, 2022. URL: https://api.semanticscholar.org/CorpusID:254234138.

[3] D. Bertillo, A. D. Donato, C. Marchetti, P. Merialdo, Enhancing accessibility of parliamentary video streams: Ai-based automatic indexing using verbatim reports, EasyChair Preprint no. 10892, EasyChair, 2023.

[4] A. Subakti, H. Murfi, N. Hariadi, The performance of bert as data representation of text clustering (2021). doi:10.21203/rs.3.rs-940164/v1.

[5] M. F. Hasani, Y. Heryadi, Y. Arifin, Lukas, W. Suparta, Density based spatial clustering of applications with noise and sentence bert embedding for indonesian utterance clustering, in: 2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE), 2023, pp. 386–391. doi:10.1109/ICCoSITE57641.2023.10127683.

[6] A. Susanto, S. Pradita, C. Stryadhi, K. Setiawan, M. Hasani, Text vectorization techniques for trending topic clustering on twitter: A comparative evaluation of tf-idf, doc2vec, and sentence-bert, 2023, pp. 1–7. doi:10.1109/ICORIS60118.2023.10352228.

[7] A. Abbas, K. Tirumala, D. Simig, S. Ganguli, A. S. Morcos, Semdedup: Data-efficient learning at web-scale through semantic deduplication, 2023. arXiv:2303.09540.

[8] T. Agnoloni, C. Marchetti, R. Battistoni, G. Briotti, Clustering similar amendments at the Italian senate, in: D. Fišer, M. Eskevich, J. Lenardič, F. de Jong (Eds.), Proceedings of the Workshop ParlaCLARIN III within the 13th Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 39–46. URL: https://aclanthology.org/2022.parlaclarin-1.7.

[9] F. Murtagh, P. Contreras, Algorithms for hierarchical clustering: an overview, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2 (2012) 86–97.

[10] H. Schütze, C. D. Manning, P. Raghavan, Introduction to information retrieval, volume 39, Cambridge University Press Cambridge, 2008.

[11] L. Hubert, P. Arabie, Comparing partitions, Journal of classification 2 (1985) 193–218.

[12] A. N. Albatineh, M. Niewiadomska-Bugaj, D. Mihalko, On similarity indices and correction for chance agreement, Journal of classification 23 (2006) 301–313.

[13] W. M. Rand, Objective criteria for the evaluation of clustering methods, Journal of the American Statistical association 66 (1971) 846–850.

[14] N. X. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: is a correction for chance necessary?, in: Proceedings of the 26th annual international conference on machine learning, 2009, pp. 1073–1080.

[15] S. Romano, J. Bailey, V. Nguyen, K. Verspoor, Standardized mutual information for clustering comparisons: one step further in adjustment for chance, in: International conference on machine learning, PMLR, 2014, pp. 1143–1151.

[16] S. Romano, N. X. Vinh, J. Bailey, K. Verspoor, Adjusting for chance clustering comparison measures, Journal of Machine Learning Research 17 (2016) 1–32. URL: http://jmlr.org/papers/v17/15-627.html.

[17] Istituto di Informatica Giuridica e Sistemi Giudiziari (IGSG) del Consiglio Nazionale delle Ricerche (CNR), Linkoln, https://linkoln.gitlab.io/, 2023.

[18] A. Peña, A. Morales, J. Fierrez, I. Serna, J. Ortega-Garcia, u. Puente, J. Córdova, G. Córdova, Leveraging Large Language Models for Topic Classification in the Domain of Public Affairs, Springer Nature Switzerland, 2023, p. 20–33. URL: http://dx.doi.org/10.1007/

978-3-031-41498-5_2. doi:10.1007/978-3-031-41498-5_2.

[19] S. Gao, L. Gao, Q. Li, J. Xu, Application of large language model in intelligent q&a of digital government, in: Proceedings of the 2023 2nd International Conference on Networks, Communications and Information Technology, CNCIT '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 24–27. URL: https://doi.org/10.1145/3605801.3605806. doi:10.1145/3605801.3605806.