

Clustering Similar Amendments at the Italian Senate

Tommaso Agnoloni¹, Carlo Marchetti², Roberto Battistoni², Giuseppe Briotti²

¹Institute of Legal Informatics and Judicial Systems (CNR-IGSG); ²Senato della Repubblica

tommaso.agnoloni@igsg.cnr.it

{carlo.marchetti, roberto.battistoni, giuseppe.briotti}@senato.it

Abstract

In this paper we describe an experiment for the application of text clustering techniques to dossiers of amendments to proposed legislation discussed in the Italian Senate. The aim is to assist the Senate staff in the detection of groups of amendments similar in their textual formulation in order to schedule their simultaneous voting. Experiments show that the exploitation (extraction, annotation and normalization) of domain features is crucial to improve the clustering performance in many problematic cases not properly dealt with by standard approaches. The similarity engine was implemented and integrated as an experimental feature in the internal application used for the management of amendments in the Senate Assembly and Committees. Thanks to the Open Data strategy pursued by the Senate for several years, all documents and data produced by the institution are publicly available for reuse in open formats.

Keywords: bills, amendments, text similarity, near-duplicates detection, clustering

1. Background and Motivation

As part of its daily activities the staff of the Italian Senate collects and organizes amendments presented by Senators on proposed laws assigned for discussion to Parliamentary Committees or for plenary discussion in the Assembly.

The Information Technology Office of the Senate develops and provides document management automation tools to speed up the process and improve the service. Application needs include assisting the operator in identifying similar amendments in a dossier in order to group them for simultaneous voting. Similar amendments can be scattered along the dossier and include those applying the same modification to different parts of the law.

Ideally, similar amendments are those producing the same effect on a proposed law. In practice, similar amendments are near duplicate texts differing in a few words in their formulation.

2. Amendments in the Legislative Process

In the lawmaking process, amendments are proposals for modification of the text of a bill, *i.e.* a proposed law under discussion within (a branch of) the Parliament. They contain proposals to change, remove or add to the existing wording of bills in order to modify their effect, allowing for bills to be improved or altered as they progress through the Parliament. Amendments are submitted in writing, to the Committee and/or to the Assembly, by the individual Senators, by the Committee that examined the bill in the referring seat, by the rapporteur or by the Government and are usually printed and distributed at the beginning of the discussion. The President decides whether they are feasible (*i.e.* related to the subject) and admissible (*i.e.* having a real modifying effect and not in contrast with resolutions

already adopted). Amendments examining and voting proceed according to a precise order, starting with those that make the most radical changes to the original text, gradually reaching those that are less distant from it. Moreover, proposals of similar content must be placed and discussed simultaneously, if possible. Amendments to an amendment may also be tabled, so-called sub-amendments, which must be voted on before the amendment itself. When voting on amendments, some of them may be absorbed (when the meaning of the amendment is included in the broader meaning of another amendment already voted and approved) or precluded (when the amendment conflicts with amendments already approved). Members of parliament can then decide to support or oppose the amendment when it is time to vote. Amendments do not need to be passed to have an effect. Non-government amendments may be proposed for other reasons: to make a political point MPs, particularly those from opposition parties, may propose amendments with the aim of advertising alternative policies or challenging the Government. These will often have little chance of succeeding but are a means of debating concerns in Parliament. Obstructionist technique (to propose a huge number of amendments differing in few words) sometimes practiced by oppositions in order to slow down the legislative process, is one of the most notable case where the automated analysis of amendment content and similarity detection would ease the work of the Senate staff.

3. Open Documents Dataset

Since 2016 the Senate of the Republic publishes all legislative documents in standard Akoma Ntoso XML format¹ with Open license CC BY 3.0. Documents are timely published via automated scripts in the

¹<http://www.akomantoso.org/>

GitHub repository `AkomaNtosoBulkData`², (Senato, 2016). This makes it easier to massively download texts for researchers, journalists, or anyone interested in accessing them automatically.

This is part of the wider Open Data strategy pursued by the Italian Senate through its data portal `dati.senato.it`³. The main purpose of such project is to make available, in open and freely reusable formats, most of the data already published on the institutional website of the Senate⁴ concerning every aspect of the political and institutional activity: bills with their process, electronic voting of the Assembly, Committees, Parliamentary Groups, Senators. This in order to ensure greater transparency on the work of the institution and encourage the concrete participation of citizens in the decision-making process.

In the `AkomaNtosoBulkData` document repository, data are structured following the same logical organization of the Senate website: for every Legislative term every bill has its own web page named "*Scheda DDL*" where it is possible to view the parliamentary phases with all related documents (presented and approved bills, reports, amendments, etc.)

The first level of the bulk data is composed of the Legislative terms. Any of them contains folders of bills in the Italian Senate. These folders contain the bills' text organized by type : proposed, debate, approved. More in detail each folder contains:

ddlpres: the text of the proposed bill or transmitted from the other branch of the Parliament;

ddlcomm: the text of the bill proposed by the Committee;

ddlmess: the text of the bill approved by the Italian Senate;

emend: the amendments discussed in the Assembly;

emendc: the amendments discussed in the Committees.

In this experimentation we focused on the *emendc* dataset of amendments presented and voted in the Committees. Amendments presented in the committees for the modification of a bill are collected in *dossiers*. Amendments are grouped by article of the bill they aim to modify. The information on the affected article is available among the amendment's metadata but not reported in its text. Metadata in the `Akoma Ntoso` structuring include signatories of the proposed amendment linked via persistent URIs to RDF metadata in the Open Data portal. The amendment content is structured in HTML for presentation on the website. For the

²[https://github.com/](https://github.com/SenatoDellaRepubblica/AkomaNtosoBulkData)

`SenatoDellaRepubblica/AkomaNtosoBulkData`

³<http://dati.senato.it>

⁴<http://www.senato.it/>

purpose of similarity analysis amendments are treated as plain text.

4. Document Similarity and Clustering

The problem of clustering (grouping by similarity) is a classical problem studied extensively in the scientific literature in statistics and data analysis (Leskovec et al., 2020), (Manning et al., 2008).

The study of the clustering problem precedes its application to the textual domain. Traditional methods for clustering have generally focused on the case of quantitative data.

In a nutshell, any document clustering approach requires a vector representation of texts with features selection and weighting, the choice of a similarity metric between pairs of vectors, and a clustering strategy.

There are different types of clustering algorithms which differ in the strategy followed to group the elements and in the various a priori assumptions (Leskovec et al., 2020). The choice of which approach to adopt depends on the characteristics of the problem under consideration.

In our case, the goal is to obtain a *partial clustering* of the elements (amendments in a *dossier*). In fact, not all amendments must be included in a cluster, but only those that have at least one "*similar*".

Furthermore, the clusters we aim to must be composed of elements that are very close to each other in their textual formulation (near duplicate texts) and not, for example, of texts that simply deal with the same topic. Our main goal at this stage is therefore to assess *lexical similarity* among texts rather than their *semantic similarity*. Moreover, in our case the number of clusters to be created is not known a priori and depends on the characteristics of the amendments in the dossier under examination. Finally, the algorithm must not be based on any a priori information or manually annotated dataset but only on the analysis of the elements (unsupervised approach).

4.1. Hierarchical Agglomerative Clustering

The most appropriate approach to clustering in this scenario is Hierarchical Agglomerative Clustering (*HAC*) (Manning et al., 2008), (Aggarwal and Zhai, 2012). Its application to amendments was previously experimented in (Notarstefano, 2016). The general concept of agglomerative clustering is to iteratively group together elements on the basis of their mutual similarity. At the beginning, each element is seen as a cluster of size 1 (*singleton* cluster). Subsequently, each element is searched for its closest element according to the chosen similarity measure and they are grouped into a cluster. At the next iteration, the process is repeated between the clusters formed in the previous step and the singleton clusters. The procedure is repeated until all the elements are grouped into a single cluster.

The process of merging the elements into successive ever larger levels of clusters creates a hierarchy, typically displayed in a dendrogram. The dendrogram

shows in a tree view the order and distance of the mergers during the hierarchical clustering process. At the lowest level, leaf nodes correspond to the individual elements. Internal nodes correspond to clusters created at each iteration. When two documents or two clusters are merged, a new node is created in the tree corresponding to the largest cluster that contains them. The process ends with the creation of a single cluster that gathers all the clusters previously created and therefore contains all the documents, corresponding to the root node of the tree.

Clusters are those groups obtained by cutting the dendrogram at a certain threshold T . The elements that have not yet been merged with any cluster at the cut-off threshold will remain in their *singleton* clusters, thus giving rise to a partial clustering. Hierarchical clustering algorithms differ by the strategy to establish grouping of clusters created at each iteration (*linkage strategy*).

We chose *complete linkage* where the similarity among two clusters amounts to the similarity of their *most distant* elements. This is equivalent to choosing the pair of clusters whose merging produces a new cluster with the minimum diameter.

4.2. Parameters Configuration

In this experiment we tested typical choices for document clustering in order to establish a baseline for further more advanced configurations:

- *tokenization* of texts around typical words separators (whitespace, tabs, carriage returns) and punctuation marks;
- token normalization using the *Snowball Stemmer* for Italian;
- removal of standard *stop-words* for the Italian language (*Python NLTK stop-words*). All other textual and numerical tokens are kept in the vector representation;
- vectorization with TF (*term-frequency*) weights and L_2 normalization to account for documents of different length;
- cosine similarity as a measure of distance between vector representations of texts. Cosine similarity is normalized between 0 and 1 (identical texts). The chosen minimum similarity threshold for grouping two texts in the same cluster is 80% (0.8);
- the criterion chosen for the HAC algorithm for clusters larger than two is the *complete-linkage* described above. With the chosen configuration, the HAC algorithm produces a normalized dendrogram with distances ranging between 0 (each element in its *singleton cluster*) and 1 (a single cluster that contains all the elements). In this way,

the value on the dendrogram at the intermediate nodes represents the maximum distance between the elements that make up the clusters that are formed at each iteration. For example, with a value of the cut-off threshold T equal to 0.2, the produced clusters will be composed of elements whose mutual distance will be *at most* equal to 0.2 and therefore *at least* 80% similar (similarity ≥ 0.8);

- we indicate this value as “*cluster compactness*” and include it among the attributes of the formed clusters.

The choices for the algorithm parameters is also driven by the application scenario where we want to use our similarity engine (see Sect. 7).

In fact, in order to simplify user interaction, we don't want to use the algorithm for exploratory analysis where the user can adjust the parameters, but we want to use fixed parameters valid independent on the document corpus that the algorithm is applied to (the dossier of amendments in our case). In particular we aim to a fixed cut-off threshold.

This is the reason why we chose TF vectorization and not TF.IDF. The IDF component of TF.IDF weighting in fact, introduces a dependency of the document vectors on the corpus and therefore a dependency on the corpus of their distances and ultimately of the cut-off threshold. Moreover, experiments using TF.IDF weighting did not show a significant performance improvement, particularly in the problematic cases (Sect. 5.3).

For the same reason of portability among different dossiers without further tuning, we chose the *complete-linkage* criterion which gives an easily interpretable cluster distance as the pairwise-distance of their most distant elements. The fixed minimum pairwise similarity threshold of 0.8, empirically established as optimal, corresponds to a 0.2 dendrogram cut-off threshold in the complete linkage case.

HAC is not very computationally efficient, as it requires at least comparing each pair of texts and doing it several times later with the resulting groups. In its most efficient implementation using *priority queues* the computational complexity is $O(N^2 \text{Log} N)$.

This type of algorithm is therefore not applicable to large datasets. In our case this is not a problem since the size of the dataset is relatively small (in the order of thousands of amendments per dossier).

5. Experiments and Evaluation

The algorithm was tested and evaluated on two dossiers, respectively:

- the dossier relating to Senate Act n. 1248⁵ composed of 1247 amendments treated in the 8th

⁵https://www.senato.it/leg/18/BGT/Schede/Ddliter/testi/51685_testi.htm

Committee (Public works, communications) and 13th (Territory, environment, environmental assets). The presented text of the amended bill is made up of 30 articles.

- the dossier relating to Senate Act n. 2272⁶ composed of 659 amendments treated in the 1st Committee (Constitutional Affairs) and 2nd (Justice). The presented text of the amended bill is made up of 19 articles.

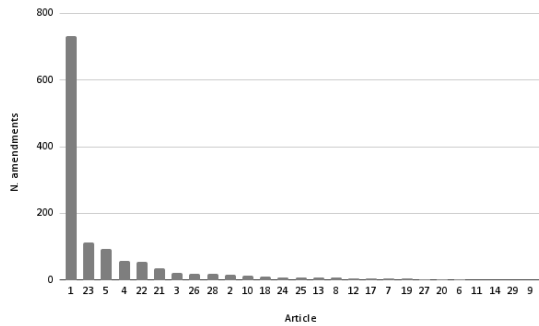


Figure 1: Act n. 1248 - distribution of number of amendments for bill article.

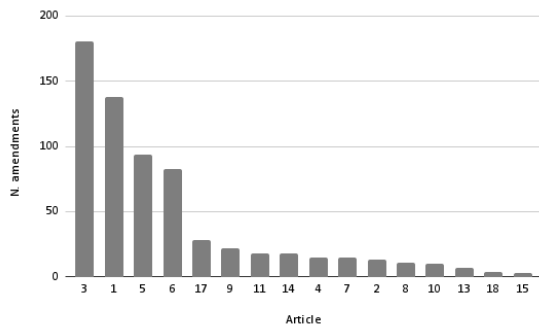


Figure 2: Act n. 2272 - distribution of number of amendments for bill article.

5.1. “Gold Standard”

A dataset with the expected “real” clustering was produced by the committees’ staff for the selected dossiers. For each of the two dossiers, each amendment was manually annotated with the *label* of the cluster to which it should be assigned or with no label in the case of an amendment without similar (*singleton*). For the dossier of Act n. 1248, manual labeling produces groupings with the following characteristics:

- 485 of the 1247 amendments (39%) have no similar and are not clustered (*singleton*);

- 762 of the 1247 amendments (61%) are grouped into 266 clusters (of size greater than or equal to 2). The minimum cluster size is 2, the maximum size is 12, the average size is 2.8. In fact, most clusters (82%) have size 2 (56%) or 3 (26%).

For the dossier of Act n. 2272:

- 246 of the 659 amendments (37%) have no similar and are not clustered (*singleton*);
- 413 of the 659 (63%) amendments are grouped into 139 clusters (of size greater than or equal to 2). The minimum cluster size is 2, the maximum size is 13, the average size is approximately 3 (2.97). In fact, most clusters (74%) have size 2 (54%) or 3 (20%).

The validation and evaluation process is in general the most difficult part of the application of a clustering algorithm since it is not generally possible to define a single “real” clustering (in principle it is correct to merge similar elements either in several small homogeneous clusters or in a single less homogeneous cluster).

5.2. Evaluation

There are several metrics used to measure the agreement between two clusterizations. The most common are ARI (Adjusted Rand Index) and AMI (Adjusted Mutual Information).

RI (Rand Index) can be seen as a percentage of correct decisions made by the algorithm. It can be calculated using the formula:

$$RI = \frac{TP+TN}{TP+FP+FN+TN}$$

The ARI variant measures the similarity between the assignment of elements to clusters provided by the algorithm and the real one ignoring the permutations and normalized with respect to random assignment (for the random assignment of elements to clusters the value of ARI is 0).

AMI is based on Shannon’s Information Theory and measures the MI (*Mutual Information*) of the algorithm assignments and the “real” ones always normalized with respect to the hypothesis of random assignment (Adjusted for Chance). AMI is equal to 1 when the two partitions are identical and is equal to 0 when the MI between two partitions is equal to the expected value for the random assignment.

Which is the most correct measure to use for the comparison between two clusterizations is an open problem. The rule of thumb (Romano et al., 2016) is:

- Use ARI when “real” clustering is made of large, homogeneously sized clusters.
- Use AMI when “real” clustering is unbalanced and there are small clusters.

⁶https://www.senato.it/leg/18/BGT/Schede/Ddliter/testi/54162_testi.htm

Being in the second case (unbalanced clustering and small clusters) we will prefer AMI to evaluate the agreement between the cluster assignment proposed by the algorithm and the “real” one, but both measures will be reported.

With the configuration of the *HAC* algorithm described above the comparison between the clusters produced by the algorithm and the “real” ones produces the AMI and ARI scores reported in Table 1

	AMI	ARI
Act n. 1248	0.71476	0.34511
Act n. 2272	0.95248	0.95469

Table 1: Evaluation against gold with AMI and ARI scores. Algorithm configuration: TF vectorization *cosine distance*; *cut-off* T=0.2; *complete linkage*.

5.3. Error Analysis

While for Act n. 2272 there is a good agreement between the grouping produced by the algorithm and the “gold” clusters (about 95%), for Act n. 1248 the results are not as good. An analysis of the assignment errors, in particular for Act no. 1248, reveals that a significant part of the wrong assignments concern amendments whose texts only differ in the identification of the subdivision affected by the modification (for example, suppression of letters or numbers):

Cluster E.1

Al comma 1, sopprimere la lettera s).
 Al comma 1, lettera s), sopprimere il numero 1).
 Al comma 1, lettera s) sopprimere il numero 1).
 Al comma 1, lettera s), sopprimere il numero 2).
 Al comma 1, lettera s), sopprimere il numero 3).
 ...
 Al comma 1, lettera s) , sopprimere il numero 4).

Another source of errors relates to amendments dealing with the deletion of an entire article . In fact, the information on the article affected by the suppression is external and is not part of the text.

Cluster E.2

Sopprimere l’articolo.
 Sopprimere l’articolo.
 Sopprimere l’articolo.
 Sopprimere l’articolo.
 ...

In all previous cases, the limit of a purely lexical comparison among texts is evident. In fact, texts are actually almost identical from the lexical point of view but the meaning and effect of the modifications are completely different.

The better evaluation scores obtained for Act n. 2272 is actually due to the almost complete absence, in the relative dossier, of these types of amendments.

Other sources of error concern the similarity between larger texts and contained texts (marked as similar in the *gold* but not always captured by the similarity measure used with the chosen threshold).

For example:

Cluster E.3

Al comma 3, apportare le seguenti modificazioni: a) sopprimere le parole: «possono essere abilitati ad assumere direttamente le funzioni di stazione appaltante e»; b) sostituire le parole: «in deroga alle disposizioni di legge in materia di contratti pubblici, fatto salvo il» con le seguenti: «nel rispetto delle disposizioni di legge in materia di contratti pubblici e nel».

Al comma 3, sostituire le parole: «e operano in deroga alle disposizioni di legge in materia di contratti pubblici, fatto salvo il rispetto» con le seguenti: «e operano nel rispetto delle disposizioni di legge in materia di contratti pubblici e».

There are other sources of error, less systematic, generally due to similarities that are difficult to grasp automatically, at least with the lexical measures used.

6. Exploiting Domain Features

Amendments are actually a very peculiar kind of technical and domain specific text. They are required to express not only the type (suppression, insertion, replacement) and the content of the modification to apply, but also to identify as accurately as possible the structural division of the bill (paragraph, letter, number..) where to apply it.

As seen in the error analysis in previous section, textual citations to legislative subdivisions are among the major sources of similarity errors when treated purely lexically. For this reason we experimented how the pre-processing of texts with the annotation and normalization of legislative citations affects the clustering performance .

We applied *Lincoln*⁷, (IGSG-CNR, 2018), a tool we previously developed for the automatic detection and linking of legal references contained in legal texts written in Italian (Bacci et al., 2019). *Lincoln* is able to detect references to entire acts, and hierarchical divisions therein, including multiple references.

6.1. Experiments with Domain Features Annotation

The following pre-annotations of texts in input to the clustering algorithm were tested and evaluated:

- **artemd** - an indivisible token (e.g. *ARTEMDI*) is added to the text in order to include the information on the amended article (information available among the metadata of the amendment);

⁷<https://gitlab.com/IGSG/LINKOLN/linkoln>

- **div** - texts are pre-processed (via a customization of the *Lincoln* annotation pipeline) in order to detect and normalize citations to legislative subdivisions. The text of the citation is replaced by an indivisible normalized token, e.g.:

Al comma 1, lettera a), numero 2), sostituire..

—→

Al DIVCOMILETAITEM2, sostituire..

- **urn** - texts are pre-processed (via *Lincoln*) in order to recognize and normalize legislative citations. The text of citations is replaced by an indivisible normalized token derived from the *urn* standard identifier (Spinosa et al., 2022) of the detected reference, e.g.:

Dopo il comma 5, inserire il seguente: « 5-bis. L'articolo 1, comma 166, della legge 30 dicembre 2018, n. 145, è sostituito dal seguente: "A valere sui contingente di personale...

—→

Dopo il DIVCOM5, inserire il seguente: «5-bis. L'STATOLEGGE20181230145ARTICOM166, è sostituito dal seguente: "A valere sui contingente di personale...)

The idea is that the replacement of citations (either to legislative acts or subdivisions) with a single normalized token allows to reduce the noise and the ambiguity in the comparison of texts.

type	annotation	AMI	ARI
0	no-annotation	0.71476	0.34511
1	artemd	0.71394	0.33288
2	artemd-div	0.85999	0.77947
3	div	0.87381	0.83304
4	urn-div	0.87325	0.83073
5	(full) artemd-urn-div	0.87069	0.81691

Table 2: Act n. 1248 - clustering evaluation with pre-annotations.

type	annotation	AMI	ARI
0	no-annotation	0.95248	0.95469
1	artemd	0.95131	0.95713
2	artemd-div	0.94706	0.95151
3	div	0.94969	0.95431
4	urn-div	0.94138	0.94576
5	(full) artemd-urn-div	0.94024	0.94381

Table 3: Act n. 2272 - clustering evaluation with pre-annotations.

Tables 2 and 3 show the results of the experiments with different pre-annotation configurations:

Type 0: (*no-annotation*) no pre-annotation of the texts;

Type 1: (*artemd*) - a token is added to the text indicating the article of the bill that is affected by the amendment;

Type 2: (*artemd-div*) - like Type 1 plus replacement of detected legislative subdivisions with normalized token;

Type 3: (*div*) - like Type 2 but without adding the token indicating the article being amended;

Type 4: (*urn-div*) - in addition to legislative subdivisions, legislative citations are detected and replaced with a normalized token derived from their *urn* standard identifier;

Type 5: (*artemd-urn-div*) texts are pre-annotated with all *features* (amended article, subdivisions and normalized legislative citations).

Results show a significant improvement (up to 16% in AMI) in clustering performance with pre-annotations of type 2 to 5 over the purely lexical tokenization (type 0). In the overall evaluation on the entire dossier, type 1 annotation does not improve the evaluation scores but in practice, it solves the issue for wrong clusters like Cluster E.2 reported in sect. 5.3

The improvement of evaluation results for Act n. 1248 only, can be explained by the fact that Act n. 2272 includes only few amendments whose textual content is mainly made of legislative references (e.g. suppressive of entire subdivisions) as also shown by the fact that clustering performance without annotation is already high. When dealing with noisy textual content introduced by the ambiguous and repetitive textual tokens of legislative citations, reference annotation and normalization has a beneficial effect in reducing wrong similarities while being neutral in all other cases.

7. Integration with the Amendments Management Application

Along their workflow, amendments are managed by Senate clerks within the application *Gestore Emendamenti* (GEM), an amendments management system developed by the IT office of the Senate.

*Similis*⁸, the service for clustering of similar amendments, was recently integrated in production as an additional experimental functionality within the GEM application.

The algorithm for similarity analysis and clustering is implemented in *Python 3.9* using the well-known *Nltk* and *SciKit* libraries. This algorithm is then included in a microservice also implemented in *Python* using the *Flask* library and exposed with a ReST interface and

⁸<https://github.com/SenatoDellaRepubblica/Similis>

JSON input/output data format. The microservice documentation is in the *OpenApi* standard.

The new functionality allows to compute the similarity clusters of a dossier of amendments and to obtain a visualization of the cluster they belong to in a column of the amendment display grid in GEM. Fig. 3 shows part of the complete dossier of amendments to Act n. 2448 of the 18th legislature (about 6665 amendments) discussed in the 5th Permanent Committee.

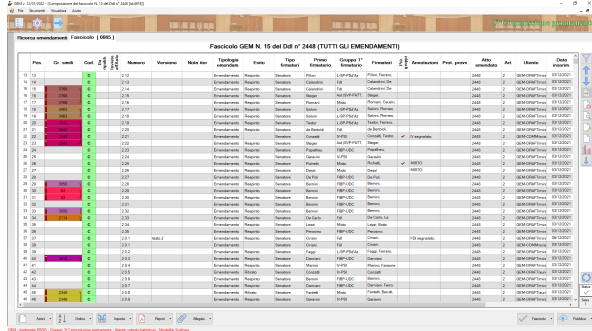


Figure 3: *GEM - amendments management application with cluster visualization.*

In the new "Similar Groups" column of the grid view (Fig. 4), for each amendment belonging to a cluster it is shown:

- the cluster ID with a unique cell color background assigned to the cluster;
- a compactness indicator (showing how close the elements of the cluster are to identical) displayed as a white bar with a shade of red (as a percentage);
- a symbol, on the right of the cell, indicating whether the amendment is at the beginning, in the middle, or at the end of the cluster in the column representation;
- a contextual menu which allows to navigate the elements of the cluster, especially useful for clusters scattered in the column.

28		C	2.27
29	3050	C	2.28
30	83	Cluster n. 3050 di 4 elementi [84%]	
31	83	C	2.30
32		C	2.31
33	3050	C	2.32
34	2114	C	2.33
35		C	2.34

Figure 4: *GEM - detailed view of the cluster column visualization.*

It is also possible to apply a filter to each cluster in order to show in the dossier grid only the amendments belonging to it.

8. Conclusions and Future Work

We experimented standard lexical similarity measures and document clustering algorithms on dossiers of legislative amendments. Preliminary results show that the extraction and annotation of domain features, in particular legislative citations within texts, allow to significantly improve the performance evaluation against manually annotated cluster assignments.

We provided an implementation of the clustering engine exposed as an internal web service within the Italian Senate IT infrastructure. The service is invoked from the application for the management of amendments in use for the Committees' and Assembly's activities. The User Interface of the application was evolved in order to include functionalities for the detection, visualization and navigation of clusters of similar amendments in the examined dossiers. The new functionality is now implemented in production and ready to be made available as an experimental feature to Senate clerks for testing and feedbacks.

Legislative amendments are a peculiar type of text, constrained by drafting rules and having several structural properties and domain features. We plan to automatically extract more of such features in order to further experiment and evaluate the effects of integrating domain knowledge in their automatic similarity analysis.

By making available this experimental functionality to final users we expect to gain a more in-depth evaluation of the quality of detected clusters, report of problematic cases and an overall evaluation of the user experience, including the effectiveness of the visualization in the User Interface, when dealing with incoming amendments dossiers on new proposed laws.

9. Acknowledgements

The authors wish to thank Paola Di Marco, Lucia Pasquini and Stefano Marci for their support in building the gold standard and for the testing efforts of Similis in the Italian Senate committees.

Bibliographical References

- Aggarwal, C. C. and Zhai, C. (2012). A Survey of Text Clustering Algorithms. In Charu C. Aggarwal et al., editors, *Mining Text Data*, pages 77–128. Springer US, Boston, MA.
- Bacci, L., Agnoloni, T., Marchetti, C., and Battistoni, R. (2019). Improving Public Access to Legislation Through Legal Citations Detection: The Lincoln Project at the Italian Senate. *Knowledge of the Law in the Big Data Age*, pages 149–158. Publisher: IOS Press.
- Leskovec, J., Rajaraman, A., and Ullman, J. D. (2020). Chapter 7 - Clustering. In *Mining of Massive Datasets*. Cambridge University Press, New York, NY, 3^o edizione edition, January.

- Manning, C. D., Schütze, H., and Raghavan, P. (2008). Hierarchical clustering. In *Introduction to Information Retrieval*, pages 346–368. Cambridge University Press, Cambridge.
- Notarstefano, J. (2016). Automated Clustering of Similar Amendments. Cern IT Lightning Talks: session 10, Jun.
- Romano, S., Vinh, N. X., Bailey, J., and Verspoor, K. (2016). Adjusting for Chance Clustering Comparison Measures. *Journal of Machine Learning Research*, (17):1–32, August.
- Spinosa, P., Francesconi, E., and Lupo, C. (2022). A Uniform Resource Name (URN) Namespace for Sources of Law (LEX). Internet Draft draft-spinosa-urn-lex-15, Internet Engineering Task Force, March. Num Pages: 56.

Language Resource References

- IGSG-CNR. (2018). *LINKOLN 2.0: the software for the automatic detection and linking of legal references contained in legal texts written in Italian*. <https://gitlab.com/IGSG/LINKOLN/linkoln>.
- Senato. (2016). *Akoma Ntoso - Bulk Data - Senato della Repubblica*. <https://github.com/SenatoDellaRepubblica/AkomaNtosoBulkData>.