HSE University
LORIA, INRIA, Univ. de Lorraine

September 12, 2024

Document Classification via Stable Graph Patterns and Conceptual AMR Graphs

International Joint Conference on Conceptual Knowledge Structures (CONCEPTS) 2024, Cádiz, Spain

Eric George Parakal, Egor Dudyrev, Sergei O. Kuznetsov and Amedeo Napoli

Introduction

- Pattern structures is an extension of Formal Concept Analysis (FCA) that enables the use of complex object descriptions. Pattern structures thus help to reduce the information loss that occurs during conceptual scaling of many-valued contexts in FCA.
- Abstract Meaning Representation (AMR) graphs ensure that texts with the same meaning have graphs with similar topology. Consequently, AMR graphs are the best document-to-graph conversion method for working with pattern structures.
- The gSOFIA algorithm mines subsample-stable graph patterns from graphs using projection-antimonotonicity and the $\Delta-$measure.
- A versatile aggregate rule classifier uses the mined stable graph patterns to classify documents in an explainable and conceptual manner. The aggregate rule classifier can operate in three distinct modes and use seven different types of penalty functions.

Pattern Structures

Let G be some set of objects, then let $(D, \sqcap)$ be a meet-semilattice of potential object descriptions and let $\delta : G \to D$ be a mapping. Then $(G, \underline{D}, \delta)$, where $\underline{D} = (D, \sqcap)$; is called a pattern structure[1] provided that the set

$$\delta(G) := \{\delta(g) \mid g \in G\}$$

generates a complete subsemilattice $(D_\delta, \sqcap)$ of $(D, \sqcap)$, i.e., every subset X of $\delta(G)$ has an infimum $\sqcap X$ in $(D, \sqcap)$ and D_δ is the set of these infima.

[1] Ganter and Kuznetsov, "Pattern Structures and Their Projections" (2001).

Pattern Concepts

If $(G, \underline{D}, \delta)$ is a pattern structure, the derivation operator is defined as

$$A^{\diamond} := \bigsqcap_{g \in A} \delta(g) \ \text{ for all } \ A \subseteq G$$

Correspondingly, for a pattern d, the derivation operator is defined as

$$d^{\diamond} := \{g \in G \mid d \sqsubseteq \delta(g)\} \ \text{ for all } \ d \in D$$

The elements of D are called patterns and are ordered as

$$c \sqsubseteq d :\iff c \sqcap d = c$$

which is called the subsumption order, where d subsumes c.

A pattern concept of $(G, \underline{D}, \delta)$ is a pair (A, d) satisfying

$$A \subseteq G,\ d \in D, \ \text{ such that } \ A^{\diamond} = d \ \text{ and } \ A = d^{\diamond}$$

The set of all pattern concepts forms the pattern concept lattice $\mathcal{L}$.

Projections

A projection ψ of a pattern set D is a mapping $\psi\colon D \to D$ on the partial order $(D, \sqsubseteq)$, which is a kernel (interior) operator, i.e., ψ is defined as

$$
\begin{aligned}
\text{monotone} \;\; & ((x \sqsubseteq y) \to (\psi(x) \sqsubseteq \psi(y))), \\
\text{contractive} \;\; & (\psi(x) \sqsubseteq x) \;\; \text{and} \\
\text{idempotent} \;\; & (\psi(\psi(x)) = \psi(x))
\end{aligned}
$$

For a pattern concept (A, d), $\psi(d)^\diamond = A$.

Example of a Pattern Structure, a Pattern Concept and a Projection

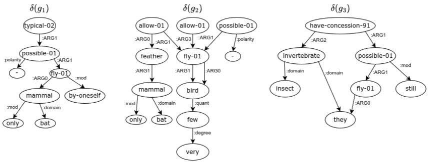

Figure 1: There are three objects (documents) g_1, g_2 and g_3, with their respective graph descriptions $\delta(g_1), \delta(g_2)$ and $\delta(g_3)$.

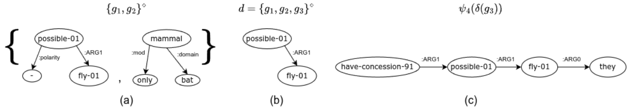

Figure 2: (a) The graph pattern concept intent $\{g_1, g_2\}^\diamond$ is obtained by intersecting the graph descriptions of the documents g_1 and g_2, here $\{g_1, g_2\}^\diamond \not\sqsubseteq \delta(g_3)$, so $\{g_1, g_2\}^{\diamond\diamond} = \{g_1, g_2\}$ (b) The graph pattern d is subsumed by all document graph descriptions $\delta(g_1), \delta(g_2)$, and $\delta(g_3)$, so $d = \{g_1, g_2, g_3\}^\diamond$ (c) The projection ψ_4, which returns the sets of 4-paths, is applied to $\delta(g_3)$.

Concept Stability

For a pattern concept (A, d), the intentional stability[2] is the probability that d remains closed upon removing a subset of objects from A, with equal probability.

The (intentional) stability of a pattern concept (A, d) is defined as

$$Stab(A, d) = \frac{|\{C \subseteq A \mid C^{\diamond} = d\}|}{2^{|A|}}$$

A pattern concept's stability reflects the independence of its intent, from the objects in its extent. A more stable pattern concept concept is more likely to have its intent be reproduced in various subsamples of a dataset and therefore, is more likely to be found in an unrelated dataset from the same population.

[2] Kuznetsov, Obiedkov, and Roth, "Reducing the Representation Complexity of Lattice-Based Taxonomies" (2007).

The gSOFIA Algorithm for Mining Graph Patterns

- The gSOFIA algorithm[3] utilizes the stability of a pattern concept's projections as an antimonotonic constraint, in order to select the most interesting graph patterns.
- Computing the stability of a pattern concept is #P-complete, thereby requiring the use of tractable bounds for stability. If p and q are two closed graph patterns such that $p \sqsubseteq q$, $\Delta(p, q)$ is defined as $\Delta(p, q) = |p^{\diamond} \setminus q^{\diamond}|$. $Stab(p)$ can then be bounded as $1 - \sum_{q \succ p} 2^{-\Delta(p,q)} \leq Stab(p) \leq 1 - 2^{-\Delta(p,q)}$. The upper bound of stability ranks patterns in the same way $\Delta(p) = \min\limits_{q \succ p} \Delta(p, q)$ and is called the $\Delta-$measure.
- The $\Delta-$measure is anti-monotonic w.r.t. any projection, i.e., $\Delta(d) \leq \Delta(\psi(d))$. Thus, $\Delta(\psi(d)) < \theta \implies \Delta(d) < \theta$, where θ is a threshold.
- The gSOFIA algorithm iteratively expands $\psi(d)$ via its preimages in ψ to obtain potentially stable graph patterns d, only if $\Delta(\psi(d)) \geq \theta$.

[3] Buzmakov, Kuznetsov, and Napoli, "Efficient Mining of Subsample-Stable Graph Patterns" (2017).

Methodology

The working of the document classifier consists of the following processes:

- Document-to-graph conversion: The DocToGraph algorithm constructs a document graph by modifying, refining and then merging the AMR graphs of the sentences, present in a document
- Mining stable graph pattern concepts: The gSOFIA algorithm mines the document graphs of the training documents of every class, to obtain a set of stable graph pattern concepts associated with each class.
- Aggregate rule classification: The aggregate rule classifier uses the mined set of concepts of every class, to classify a test document, via the graph subsumption relation.

Document-to-Graph Conversion

- AMR graphs[4] are well suited to work with pattern structures as they maximize intra-class document graph intersection, whereas inter-class document graph intersection is minimized.
- Given a document g_i, the DocToGraph algorithm obtains its graph description $\delta(g_i)$ by modifying, refining, and then merging the set of AMR graphs $\{AMR_{ji}\}$.
- Each AMR_{ji} is generated via the AMRParser procedure for each of the sentences t_{ji}, of every g_i.
- The document graphs are not syntactically correct, pure AMR graphs. When mining for stable graph pattern concepts, the presence of variables that act as internal nodes in pure AMR graphs can result in unnecessary overlap between the graph pattern concept intents of various classes.
- The ModifyGraph algorithm replaces all the internal nodes of each AMR_{ji} with their respective instance entities, converts the node labels to lowercase, and lemmatizes them.

[4] Banarescu et al., "Abstract Meaning Representation for Sembanking" (2013).

The DocToGraph Algorithm

Algorithm 1 DocToGraph

Input: a document g_i
Output: a graph description $\delta(g_i)$

$T_i \leftarrow$ findSentences(g_i)
for all $t_{ji} \in T_i$ **do**
  $AMR_{ji} \leftarrow$ AMRParser(t_{ji})
  $MOD_{ji} \leftarrow$ ModifyGraph(AMR_{ji})
  $REF_{ji} \leftarrow$ refineGraph(MOD_{ji})
end for
$\delta(g_i) \leftarrow$ mergeGraphs($\{REF_{ji}\}$)
return $\delta(g_i)$

Example of a Document Graph

The figure below illustrates the document graph, i.e., the graph description $\delta(g_1)$ obtained from a document having the following two sentences: "A mammal typically cannot fly by itself. Bats are the only flying mammal."

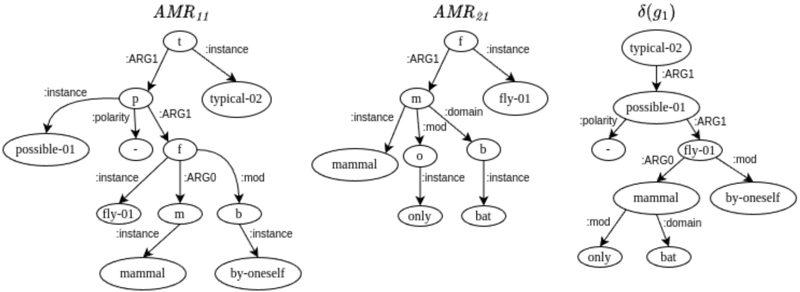

Figure 3: An example of two AMR graphs AMR_{11} and AMR_{21}, corresponding to individual sentences, which are modified, refined and then merged to obtain the graph description $\delta(g_1)$.

The ModifyGraph Algorithm

Algorithm 2 ModifyGraph

Input: an AMR graph AMR_{ji}
Output: a modified graph MOD_{ji} of AMR_{ji}

$MOD_{ji} \leftarrow \phi$
$E_j \leftarrow$ findAllEdges(AMR_{ji})
for all $e_j \in E_j$ **do**
  if relation(e_j) $\neq$ ":instance" **then**
    $MOD_{ji} \leftarrow$ addEdge(findInstance(head(e_j)), relation(e_j), findInstance(tail(e_j)))
  end if
end for
return MOD_{ji}

Mining Stable Graph Pattern Concepts

- The gSOFIA algorithm uses canonically ordered frequent graph patterns mined using the Gaston[5] algorithm to grow the Δ-stable graph patterns.
- The Gaston algorithm cannot work with directed graphs with parallel edges and self-loops.
- The ModifyGraph algorithm circumvents this restriction by using a unique triple for each edge of a modified graph MOD_{ji}. The triple encodes edge directionality and differentiates edges with the same nodes but different relations between them.
- Each graph pattern concepts lattice $\mathcal{L}_c$ belonging to a class c has graph pattern concepts whose graph patterns' Δ-stability exceeds the threshold θ.

[5] Nijssen and Kok, "The Gaston Tool for Frequent Subgraph Mining" (2004).

Aggregate Score Calculation

The aggregate rule classifier generates an aggregate score $S^{\mathcal{L}_c}$ for each class c. A test document g_{test} is classified as belonging to the class with the highest $S^{\mathcal{L}_c}$.

$$S^{\mathcal{L}_c} = \frac{1}{|\mathcal{L}_c|} \sum_{(A_c, d) \in \mathcal{L}_c} \frac{\sum_{p_i \in d} support(p_i) \times \#size(p_i)}{penalty((A_c, d))} [d \sqsubseteq \delta(g_{test})]$$

Each aggregate score is the normalized, weighted, cumulative sum of the subgraphs p_i present in those graph pattern concepts (A_c, d), belonging to the graph pattern concept lattice $\mathcal{L}_c$, whose graph pattern intents d are subsumed by the graph description of the test document $\delta(g_{test})$.

Toy Example of the Aggregate Rule Classifier

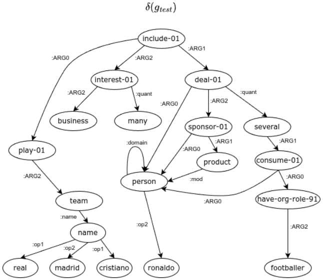

Figure 4: An example of a description of a test object g_{test}, that is to be classified as belonging to one of the three classes: *sports*, *business* or *food*.

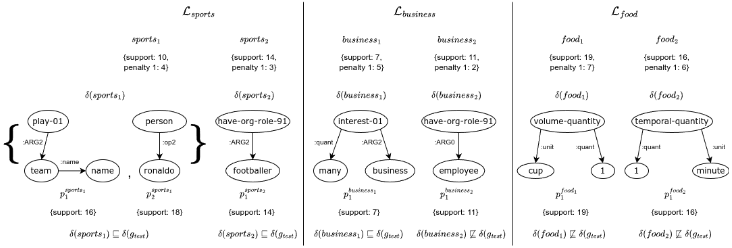

Figure 5: An example of a three concept lattices $\mathcal{L}_{sports}$, $\mathcal{L}_{business}$ and $\mathcal{L}_{food}$, along with their respective concepts that are used to classify g_{test}.

Example of Aggregate Scores Calculation

The aggregate scores of the *sports*, *business* or *food* classes for the g_{test} shown in the toy example, are calculated below.

$$S^{\mathcal{L}_{sports}} = \frac{1}{|\mathcal{L}_{sports}|}\left(\frac{support(p_1^{sports_1}) \times \#size(p_1^{sports_1}) + support(p_2^{sports_1}) \times \#size(p_2^{sports_1})}{penalty(sports_1)}\right.$$

$$\left. + \frac{support(p_1^{sports_2}) \times \#size(p_1^{sports_2})}{penalty(sports_2)}\right) = \frac{1}{2}\left(\frac{16 \times 3 + 18 \times 2}{4} + \frac{14 \times 2}{3}\right) = 15.16$$

$$S^{\mathcal{L}_{business}} = \frac{1}{|\mathcal{L}_{business}|}\left(\frac{support(p_1^{business_1}) \times \#size(p_1^{business_1})}{penalty(business_1)}\right) = \frac{1}{2}\left(\frac{7 \times 3}{5}\right) = 2.1$$

$S^{\mathcal{L}_{food}} = 0$

Since $S^{\mathcal{L}_{sports}}$ has the highest value, g_{test} is classified as belonging to the class *sports*.

Concept Penalties

A graph pattern concept belonging to class c is penalized if its graph pattern intent is subsumed by the descriptions of training documents not from its class, i.e., $C \setminus \{c\}$. There are a total of 7 different concept penalties defined as below.

$$penalty_1((A_c, d)) = |d^{\diamond} \setminus A_c| \sum_{p_i \in d} \#size(p_i)$$

$$penalty_2((A_c, d)) = \sum_{g \in d^{\diamond} \setminus A_c} \frac{1}{\#size(\delta(g))}$$

$$penalty_3((A_c, d)) = \sum_{g \in d^{\diamond} \setminus A_c} \frac{1}{|max.degree(p_i \mid p_i \in d) - max.degree(\delta(g))|}$$

Concept Penalties (continued)

$$penalty_4((A_c, d)) = \sum_{e \in edges(d)} edge\_penalty(e)$$

$$\text{where } edge\_penalty(e) = \sum_{\tilde{c} \in C} \sum_{\substack{(A_{\tilde{c}}, \tilde{d}) \in \mathcal{L}_{\tilde{c}} \\ e \in edges(\tilde{d})}} |e^{\diamond} \setminus G_{\tilde{c}}|$$

$$penalty_5((A_c, d)) = |d^{\diamond}|$$

$$penalty_6\big((A_c, d)\big) = |d^{\diamond} \setminus A_c|\,|d|$$

$$penalty_{\text{baseline}}\big((A_c, d)\big) = 1$$

Operational Modes

The previously described operation is the aggregate rule classifier's default mode of operation. To provide increased versatility, the aggregate rule classifier can also operate in the following two modes of operation:

- Frequent subgraphs mode: The frequent subgraphs of every class c, F_c are an additional output of the Gaston algorithm obtained during the operation of the gSOFIA algorithm. Each frequent subgraph $f_i^c \in F_c$ can be used in the same manner the aggregate rule classifier uses a graph pattern concept intent for classifying documents.
- Equivalence classes mode: If $\dot{E}_c$ is an equivalence class, then for any two graph patterns d_i and d_j, such that $i \neq j$ and $\{d_i, d_j\} \in \dot{E}_c$, $d_i^{\diamond\diamond} = d_j^{\diamond\diamond}$. An equivalence class is used in the same manner that the aggregate rule classifier uses a graph pattern concept concept intent for classifying documents.

Experiments and Results

- It is computationally expensive to operate the document classifier, and the benchmarking of its document classification performance requires the use of small datasets.
- The first dataset[6], referred to here as the 10 newsgroups dataset, contains 1000 documents belonging to 10 classes. The first dataset is a subset of a larger dataset consisting of newsgroup documents[7].
- The second dataset[8], referred to here as the BBC sport dataset, includes 737 documents that belong to 5 classes. These documents are obtained from the BBC Sport website and are related to sports news articles.
- The experiments results are available in the in the online repository [9].

---

[6] Baxter, *(10)Dataset Text Document Classification* (2020).

[7] Mitchell, *20 Newsgroups* (1999).

[8] Greene and Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering" (2006).

[9] https://github.com/ericparakal/stable-AMR-graphs-document-classifier

Baseline Models Performance

The aggregate rule classifier's document classification performance is compared to three baseline models. The first two models, SVM and logistic regression, are trained using the TF-IDF vectorized texts of the documents. The third model is a simple four-layer Graph Convolutional Network (GCN) that is trained to classify the document graphs.

Table 1: Table describing the document classification performances of the baseline models, in terms of the macro-averaged F_1 score.

Dataset	Logistic regression	GCN	SVM
10 newsgroups	**0.9502**	0.8478	0.8018
BBC sport	0.9732	0.8906	**0.9733**

Default Mode of Operation Performance

Table 2: Table describing the document classification performance, in terms of the macro-averaged F_1 score, of our aggregate rule classifier for all penalties, while operating in its default mode of operation.

Dataset	$penalty_{\text{baseline}}$	$penalty_1$	$penalty_2$	$penalty_3$	$penalty_4$	$penalty_5$	$penalty_6$
10 newsgroups	0.6425	0.7303	0.7004	0.7004	0.2096	0.6946	**0.7398**
BBC sport	0.7884	**0.9503**	0.8855	0.8858	0.6829	0.8732	**0.9503**

Equivalence Classes Mode of Operation Performance

Table 3: Table describing the document classification performance, in terms of the macro-averaged F_1 score, of our aggregate rule classifier for all penalties, while operating in its equivalence classes mode of operation.

Dataset	$penalty_{\text{baseline}}$	$penalty_1$	$penalty_2$	$penalty_3$	$penalty_4$	$penalty_5$	$penalty_6$
10 newsgroups	0.6714	**0.7957**	0.7735	0.7750	0.3244	0.7876	0.7739
BBC sport	0.7414	0.9227	0.9046	0.8941	0.5437	0.9084	**0.9300**

Frequent Subgraphs Mode of Operation Performance

Table 4: Table describing the document classification performance, in terms of the macro-averaged F_1 score, of our aggregate rule classifier for all penalties, while operating in its frequent subgraphs mode of operation.

Dataset	$penalty_{\text{baseline}}$	$penalty_1$	$penalty_2$	$penalty_3$	$penalty_4$	$penalty_5$	$penalty_6$
10 newsgroups	0.7412	0.8246	0.8017	0.7767	0.6266	**0.8287**	0.8021
BBC sport	0.7013	0.9300	0.9300	0.9187	**0.9436**	0.8639	0.9300

Conclusion and Future Works

- The aggregate rule classifier proved the feasibility of using graph pattern concepts for classifying documents in an explainable and conceptual manner.
- By using seven different concept penalties and three distinct modes of operation, the aggregate rule classifier can provide respectable document classification performance.
- Future works intend to increase the method's document classification performance and scalability by using stochastic explainability methods, such as concept whitening[10], instead of deterministic methods like an aggregate rule classifier.

[10] Chen, Bei, and Rudin, "Concept whitening for interpretable image recognition" (2020).

References I

- Banarescu, Laura et al. "Abstract Meaning Representation for Sembanking". In: *LAW@ACL*. The Association for Computer Linguistics, 2013, pp. 178–186.
- Baxter, Jensen. *(10)Dataset Text Document Classification*. 2020. URL: https://www.kaggle.com/datasets/jensenbaxter/10dataset-text-document-classification.
- Buzmakov, Aleksey, Sergei O. Kuznetsov, and Amedeo Napoli. "Efficient Mining of Subsample-Stable Graph Patterns". In: *ICDM*. IEEE Computer Society, 2017, pp. 757–762.
- Chen, Zhi, Yijie Bei, and Cynthia Rudin. "Concept whitening for interpretable image recognition". In: *Nat. Mach. Intell.* 2.12 (2020), pp. 772–782.
- Ganter, Bernhard and Sergei O. Kuznetsov. "Pattern Structures and Their Projections". In: *ICCS*. Vol. 2120. Lecture Notes in Computer Science. Springer, 2001, pp. 129–142.

References II

- Greene, Derek and Padraig Cunningham. "Practical solutions to the problem of diagonal dominance in kernel document clustering". In: *ICML*. Vol. 148. ACM International Conference Proceeding Series. ACM, 2006, pp. 377–384.
- Kuznetsov, Sergei O., Sergei A. Obiedkov, and Camille Roth. "Reducing the Representation Complexity of Lattice-Based Taxonomies". In: *ICCS*. Vol. 4604. Lecture Notes in Computer Science. Springer, 2007, pp. 241–254.
- Mitchell, Tom. *20 Newsgroups*. 1999. URL: http://kdd.ics.uci.edudatabases/20newsgroups/20newsgroups.html.
- Nijssen, Siegfried and Joost N. Kok. "The Gaston Tool for Frequent Subgraph Mining". In: *GraBaTs*. Vol. 127. Electronic Notes in Theoretical Computer Science 1. Elsevier, 2004, pp. 77–87.