Evaluating Flowchart Recognition for Patent Retrieval

Mihai Lupu
Vienna University of Technology
Favoriten Strasse 9-11/188
A-1040 Vienna
lupu@ifs.tuwien.ac.at

Florina Piroi
Vienna University of Technology
Favoriten Strasse 9-11/188
A-1040 Vienna
piroi@ifs.tuwien.ac.at

Allan Hanbury
Vienna University of Technology
Favoriten Strasse 9-11/188
A-1040 Vienna
hanbury@ifs.tuwien.ac.at

ABSTRACT

A set of measures for assessing the effectiveness of flowchart recognition methods in the context of patent-related use cases are presented. Two perspectives on the task are envisaged: a traditional, re-use of bitmap flowcharts use-case and a search-related use-case. A graph topology-based measure is analyzed for the first and a particular version of precision/recall for the second. We find that the graph-based measure has a higher discriminating power, but comes at higher computational costs than the search-based measures. The evaluation of the runs in the absence of ground truth is also investigated and found to provide comparable results if runs from the same group are not allowed to unbalance the synthetically generated truth sets.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.4.9 [**Image Processing and Computer Vision**]: Applications; H.3.4 [**Systems and Software**]: Performance evaluation

General Terms

Experimentation,Measurement,Performance

Keywords

Flowchart, patent, evaluation

1. INTRODUCTION

In the context of the CLEF-IP evaluation campaign, we are interested in all forms of information access for the purposes of the so-called *search for innovation* use case domain, which, for all practical purposes of this campaign, focuses on the issue of accessing patent information. While the main focus of the campaign, as well as of other campaigns working with patent data such as the NTCIR [4] or TREC [8], is on text data, patent searchers must, particularly for some technology areas, rely primarily on information present in images [9].

Among image processing tasks of the patent domain, chemistry has traditionally been in the spotlight due to the immense commercial interests, technological challenges, as well as the history of the field. In this report however, we focus on a different kind of image often present in patent documents: flowcharts. The issue of providing tools to handle patent images, and the issue of methods to evaluate these tools have received more attention recently, in the context of efforts to provide multimodal and multilingual information access tools. In a previous campaign [14] a high-level retrieval task covering both images and multilingual text was attempted, but participation was limited to one participant. We can presume that this was due to the difficulty of the task. In 2012, the tasks of CLEF-IP were therefore more granular. This report is based on the 13 runs received for the flowchart recognition task, from three participating groups: UAB (Autonomous University of Barcelona) [16] (4 runs), INRIA [22] (1 run), and JRC (Joanneum Research Centre) [12] (8 runs).

The rest of the article is structured as follows: a very brief related work is presented next, followed by a description of the test collection and the use-cases which lie at the basis of the effectiveness measures discussed. These metrics are described in detail in Section 4 and experimental results using them are shown in Section 5. A method to evaluate runs in the absence of the ground truth is proposed in Section 6. Finally, conclusions and future work are listed in Section 7.

2. RELATED WORK

The task of image recognition is a research field in itself. Patent image recognition, and in particular the case of flowcharts, is significantly different in nature compared to photographic image processing [9].

In a very specific context such as that of patents, where images are generally binary black-white bitmaps, each type of image needs to be treated separately, and existing efforts have focused on chemical data [8]. The evaluation of these relies heavily on specificities of the domain and makes a set of simplifying assumptions about the images to be processed.

In the general case, flowchart recognition work concentrates on the processing of hand-drawn images [11, 6]. These works use a recently created test collection [1]. The focus of this test collection is on the low level aspects of the recognition process and provides, for 78 flowcharts only, stroke and symbol annotations. To the best of our knowledge there is no study looking into evaluating flowchart recognition from

EVIA '13 Tokyo, Japan

the point of view of a searcher.

On the other hand, there is a substantial amount of work on the development of test collections and measures in general in information retrieval [19]. In the absence of human assessments, we are left to compare the results of the proposed effectiveness measures with each other and to establish their discriminative power [17]. For the latter, we do not use a pair-wise test, but rather the recently proposed method to address the Multiple Comparisons Problem which occurs in the context of simultaneous testing of many hypotheses using a fixed set of data [3].

3. COLLECTION AND USER MODEL

The flowchart collection consists of 100 bitmap images and their corresponding graph textual representations. The set of images was selected manually from the collection of patent images used in CLEF-IP 2011 [14]. The selection aimed to make sure that 1. the images represented indeed flowcharts and 2. the flowcharts did not contain non-standard elements introduced by the patent applicant (e.g. hand annotations, meta-nodes, other images as nodes). In the final tests one image was left out of the evaluation because it contained no text and node labels are an essential part of the utility of a flowchart.

Based on the best practices developed in the context of the PROMISE[1] project [5], and in cooperation with patent users, we have defined two use-case scenarios for this task:

Inventor is a user of patent information in a research and innovation process. The user obtains a bitmap of a flowchart, either from an existing patent, scientific article, or hand-drawn and wishes to obtain a graph representation for further processing

Searcher is a user of patent information in a novelty search process [7]. The user has a general request for information (similar to common usage in text retrieval) and desires to exploit information available in flowchart images present in patent documents to identify related documents.

The two use-case scenarios are certainly related and one of the aspects that we investigate in this report is the connection and correlation between metrics developed for each of them.

The task given to participants consisted of transforming a bitmap representation of a flowchart into a graph textual representation. Participants were required to identify the nodes and how they are connected, but also the type of nodes (e.g. diamond, circle, rectangle,...), type of edges (e.g. continuous, dashed), direction of edges and text of node labels. A particular type of node is specific to this use-case: *no-box* indicates those nodes which are not actually part of the flowchart itself, but denote labels attached to the nodes of the flowchart. They are connected to their corresponding nodes by an edge whose type we denote as *wiggly*.

To solve this task, participants were provided an initial training set. Unfortunately, this came without a target metric they could optimize for because the precise details of the metrics discussed in this report were not fixed at the time.

[1]http://www.promise-noe.eu

4. EVALUATION METRICS

Given the use-case just presented, the question is how do we evaluate the effectiveness of the participating runs in recognizing the flowcharts depicted in the 99 topic images. There are two ways of approaching the problem: topology and functional similarity. The first set considers primarily the structure of the graphs and secondarily the content and type of nodes. The second attempts to evaluate the recognition from the perspective of a potential search user, by generating a set of potential queries for each flowchart. Each has intrinsic advantages and disadvantages, which we will discuss in what follows. Section 5 will then describe the experimental results obtained with both of these sets of metrics.

4.1 Graph-oriented

The graph oriented evaluation measures were initially proposed and used for the CLEF-IP 2012 evaluation campaign. Primarily, the score of a submitted topic result R was computed based on the size of the maximal common subgraph (MCS) between it and the manually created graph representation of the topic, G as follows:

$$score(R,G) = \frac{|mcs(R,G)|}{|R| + |G| - |mcs(R,G)|} \quad (1)$$

where $|\cdot|$ denotes the size of the graph and $mcs(\cdot,\cdot)$ denotes the maximal common subgraph of two graphs.

This scoring function was proposed to be calculated at three levels [13]:

basic - using solely the structure of the graph

intermediate - matching node types

complete - matching node types and text labels

The initial interpretation of these measures was inspired by work in the chemical domain, where chemical formulas can also be represented as graphs and common sub-formulas are identified as maximal common subgraphs [15]. The strict interpretation prohibits the node matching function defining the MCS to link nodes of different types. This may be very appropriate for chemistry, but for our case we found this to be too strict, particularly since flowcharts use the shapes of nodes more as indications rather than hard rules. Consider for instance two flowcharts as in Figure 1. Flowchart B matches Flowchart A perfectly with the exception of the decision node, which was erroneously recognized as an oval. In such a situation, a strict interpretation of the *intermediate* scoring method would reduce the size of the maximal common subgraph from 8[2] nodes to only 5 (a 38% decrease) and the number of edges from 8 to 4 (a 50% decrease). The effect would be further accentuated when using the *complete* scoring method. The effects of the OCR tools applied by all participants were in initial experiments so acute that most maximal common subgraphs found were of size 1 or 2, and this approach was abandoned in favor of a more relaxed interpretation of the matching levels.

The relaxed interpretation of the node content and type matching is as follows: given the set of maximal common subgraphs (unless a perfect match is found there will often be multiple subgraphs of the same maximum size) consider

[2]There are 7 visible nodes and 1 POINT node where the edges 4-6 (d-f) and 5-6 (e-f) join

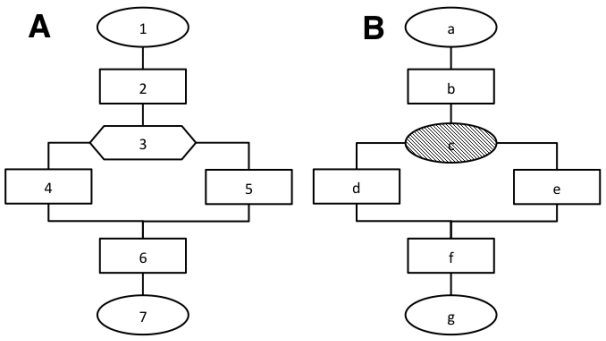


Figure 1: A common type error can have an extremely high penalty on the Maximal Common Subgraph of two flowchart graphs

the best percentage of matched node types and the minimum edit distance between all pairs of nodes. More formally, for a run graph R and given a ground-truth graph G, we define two additional measures, MTM (maximum type match) and MXM (maximum text match) as follows:

$$MTM(R,G) = \max_{M \in \mathcal{M}} \left(\frac{\sum_{x \in V(R)} \mathbf{1}(type(x) = type(f_M(x))}{|V(R)|} \right) \quad (2)$$

$$MXM(R,G) = \max_{M \in \mathcal{M}} \left(\frac{\sum_{x \in V(R)} editDist(x, f_M(x))}{\sum_{x \in V(G)} textLength(x)} \right) \quad (3)$$

where $\mathcal{M}$ is the set of maximal common subgraphs and $f_M : V(R) \rightarrow V(G)$ is the injective function matching the nodes of R to those of G, which defines the maximal common subgraph M.

The MXM is not normalized between topics. The denominator is the total number of characters in the ground truth graph, not in the run graph because that would reward a run which returns significantly less text than it should, versus a run which returns OCR with possible spelling errors.

The immediate question arising from these definitions is whether max is the best aggregation function. This might seem to favor the run and give high scores. Even though this ultimately should not be relevant for a comparative study as is the case here, one might be concerned of obtaining metrics based on cherry-picked matches. We argue that this is not the case, and that the max aggregator makes sense because ultimately the generated graph would either be used to provide a visual representation to the user (with the purpose of recombining its parts for a new invention, for instance) or to issue queries on it. In the first case, the human will immediately identify the best match visually, both from the point of view of the node types and the text labels of the nodes. The second case we discuss in the next section, but for it it makes no difference which maximal common subgraph in particular is selected.

Finally, we should note at this point that edges types and directions are never taken into account neither here nor in the following section, despite the initial requirements from the participants to identify the type, direction and labels of the edges. Our observations lead us to the conclusion that the particular nature of the edges was too unreliable for this particular evaluation exercise, but should be the focus of a different study. This "unreliability" is two-fold: first, recognition systems have a very hard time distinguishing the arrow at the end of a line; second, and most importantly, the directions are not consistently indicated in flowcharts. Very often there is an implicit direction (top-bottom, left -right) and only some edges have specific direction markers. Even more, where they exist, direction markers are potentially placed anywhere on the edge.

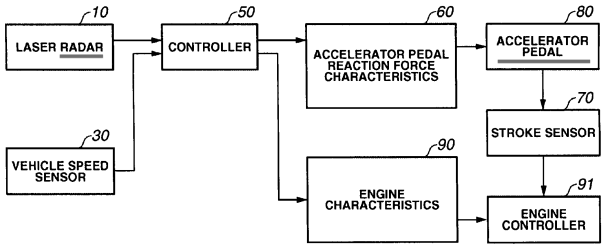


Figure 2: A possible query modality would look for connected nodes.

4.2 Query-oriented

Before we begin, we should note that we will be using the term *"query"* in what follows to indicate a potential query that a user might have for a collection of flowcharts. We will refer to a flowchart image as a *"topic"*. In this understanding, a topic contains, or is represented by, a set of queries, similar to the general TREC terminology, but a query here can only have a yes/no answer (whether the run can/cannot retrieve the particular flowchart image given the query) and a topic is evaluated using all the queries at the same time.

To complement the primarily topological approach just presented, we can also investigate a functional view of the flowchart recognition task. As described in Section 3, probably the main use of the results of this task would be to make the information present therein searchable using keywords. The simplest variant of this is to only look at one node. The more interesting one, and the one we consider in this report, is that where the user is interested in connected nodes. For instance, Figure 2 shows an answer to a hypothetical query asking for patents describing systems that control the acceleration pedal (presumably of a car) with a radar system. One could go even further and ask for more complex interactions. To continue the example, one may ask for patents which control the *engine* via the *accelerator pedal* based on input from a *radar*.

Such queries can be easily generated from the ground-truth representation of a given topic by selecting all pairs of nodes connected by a path. However, to keep with the use-case scenario at hand, two restrictions are imposed on the set of nodes used in this process:

1. must contain a text label
2. must not be of type no-box (i.e must be an actual node in the flowchart, not a label node)

Given this set of "eligible" nodes, we compute for each pair the minimum path between them using the familiar Dijkstra algorithm and consider the pair a query if the path length is not ∞. We do so in both directions, to cast as wide a net as possible on the possible queries for each topic.

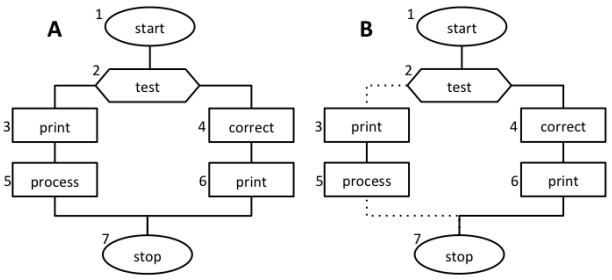


Figure 3: Duplicate labels can hide missing paths.

Having created such a set of queries, we can then compute how many of them could have been answered by the run graph (i.e. recall) by trying to find a path between two nodes with the given text labels in the run graph. If we denote by $Q \subseteq V(G) \times V(G)$ the set of queries represented by all pairs of eligible nodes of the ground truth graph G, and a function $m : V(G) \rightarrow V(R)$ which maps a node g of the ground truth graph to a node r of the run graph R based on their labels, we can formalize this recall measure in the usual way as

$$recall(R,G) = \frac{1}{|Q|}\left(\sum_{p \in Q} \mathbf{1}\left(dijk(m(p_{start}), m(p_{stop})) < \infty\right)\right) \quad (4)$$

where $dijk(x, y)$ returns the length of the minimum path between nodes x and y in graph R.

It is equally easy to compute a precision measure by changing the denominator of Equation 4. In exactly the same way as we have generated the set of queries from the ground truth graph, we can generate Q_R the set of potentially answerable queries by the run graph R, and replace $|Q|$ with the newly computed $|Q_R|$. However, this measure does not seem to have a strong foundation in the use-case. While one may argue that it is preferable that the run graph does not introduce spurious paths, such paths would never actually be searched for, since based on our observations they are often the result of misspellings due to the OCR software, and rarely due to the insertion of additional edges. Nevertheless, because of this connection with the OCR errors, it is worth investigating further to observe any connections with the MXM.

4.2.1 Duplicates

A potential issue with the above described method of generating queries is its absolute reliance on node labels in the query definition and tests. In some cases, the same label appears on multiple nodes within a flowchart. If that is the case, the way queries are generated might ignore potential errors. For instance, in Figure 3 we see the ground truth graph A and a hypothetic run graph B. The latter is missing two edges (2-3 and 5-7), resulting in two connected components instead of one. The queries generated based on the node pairs (3,4), (6,4) are identical: ("print" – "correct") and will only be tested once. For this test, we can say that node 6 hides node 3 and the missing edges will not result in a corresponding penalty (though some penalty is likely to be incurred because of the existence of nodes like node 5).

This seems indeed problematic, but considering that our success measure is whether the graph was returned or not based on a pair of terms representing labels, the lack of penalty in the case of label duplicates may be justifiable.

In practice we found the presence of duplicates to be less significant. Of all the 1926 nodes in the collection, only 48 had labels present on another node in the same graph (2.4%). We implemented the query generation method with the option to de-duplicate nodes and found that of the 5581 queries thus created, 5070 were unique (i.e. 9.1% duplicates). The difference between the percentage of duplicates in the queries and in the nodes is explained by the presence of duplicates in larger graphs, and the earlier observation that the number of queries grows exponentially in the number of nodes in a graph.

Correctly using these de-duplicated queries is however not immediately obvious, because the (dis-)connection patterns make this a combinatorial problem. Correctly identifying which de-duplicated node of the ground truth should match which one of the different options of the run graph should perhaps be the focus of another study. For now, we rely on the expected small impact of this small number of duplications.

5. EXPERIMENTS

In this section we present the experimental results of applying the measures described in the previous section to the ranking of the 13 runs submitted to CLEF-IP 2012. In this section we will show the results obtained with each measure, and, together with them an indication of the quality of the measure in terms of its ability to distinguish the different runs. For this later part we have followed the recent suggestion [3] of performing an ANOVA to test the omnibus null hypothesis that all systems are equal according to a particular measure. In all cases this was rejected, indicating that at least two runs were statistically significantly different. We have then performed post-hoc pairwise two sided t-tests to identify these significantly different pairs. As the primary purpose of this report is not to identify which is the best system, but rather whether the proposed measures and procedures can distinguish between systems, we show the significance results in a manner similar to the recent report of Sakai [18], namely by showing the set of p-values for the entire set of system pairs. Such plots will always show an x-axis with 79 values (13 runs, $(13 \cdot 12)/2$ pairs) and a y-axis marked at the 0.05 significance level.

In each of the following subsections we shall start with a description of the implementation decisions taken for the metrics at hand. A common decision was to normalize the text labels of the nodes of both the ground truth and the runs, by keeping only word symbols ([a-zA-Z0-9_]), merging white spaces where they occur together and lowercasing. We considered this to be a baseline for what a generic search engine would do in the indexing process.

5.1 Maximal Common Subgraph

To compute the maximal common subgraph we implemented the McGregor algorithm [10]. This is a well known combinatorial problem whose time complexity is $O(|V(G)|!)$ [2] and which therefore, in the worst case, is impossible to solve for some of the graphs in our collection. This is particularly so since for the MTM and MXM measures we need to have the entire set of MCSs. Unlike the original algorithm which prunes the search space on a strict inequality condition, we have to use a greater or equal condition to reach and store

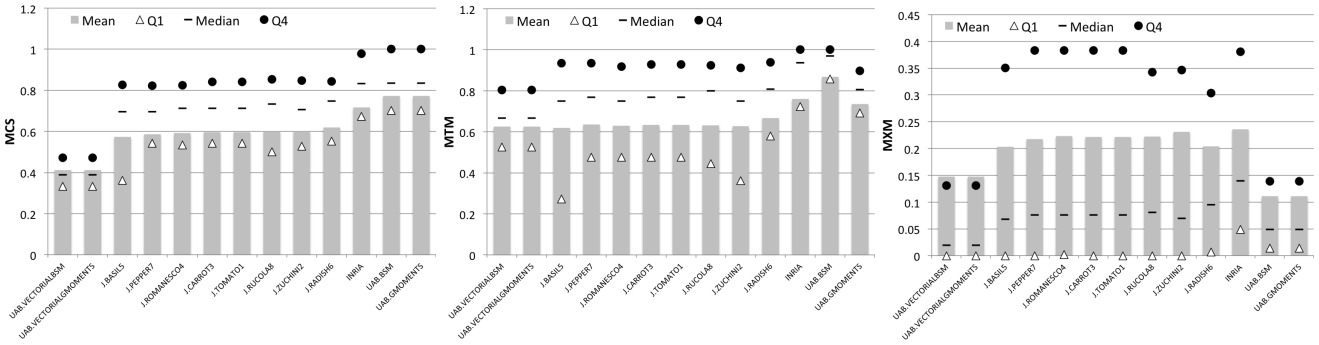


Figure 4: Graph-based measures.

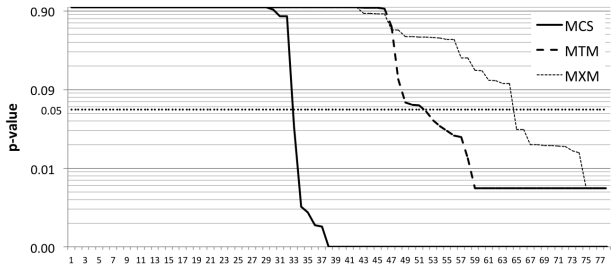


Figure 5: P-values for pairs of systems for the three graph-based measures

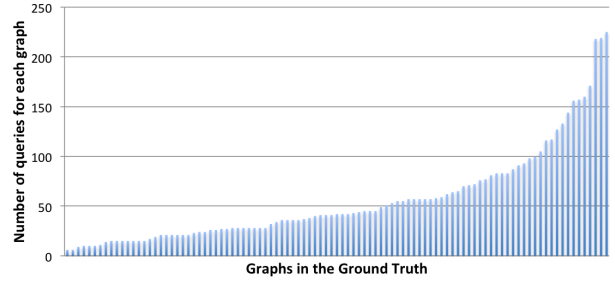


Figure 6: The number of queries per ground truth graph

all MCS of the same size.

In practice optimizations are available based on node labeling. In our case we only used a concept of anchors, as follows:

Definition 1. *For each pair of graphs (R, G), a pair of nodes $(v_R, v_G) \in V(R) \times V(G)$ are* anchored *iff $label(v_R) = label(v_G)$ and $\forall v \in V(R), label(v_R) \neq label(v)$ and $\forall v \in V(G), label(v_G) \neq label(v)$.*

This definition simply selects those pairs of nodes which are unambiguously matched between the run and ground truth graph. The anchored nodes will then reduce, sometimes significantly, the search space for the MCS. They will also potentially prevent us from finding the true MCS based on the topology alone, but such a result would not be desirable anyway, since it would clearly be a mismatch with respect to the way a human would perceive the two graphs.

Where such an anchor cannot be found for a particular node v_R of the run graph, a candidate list is created with the nodes of the ground truth graph, sorted in order of the edit distance between their labels and the label of v_R. The McGregor algorithm iterates in a backtracking way over these sets of candidates for each node.

Moving on to the results, Figure 4 plots a summary of the values obtained for each system and for each of the three measures. The three figures show the runs in increasing order of their MCS, since MTM and MXM are only complementing measures, without meaning outside of the context of MCS. The MXM, based on the edit distance, indicates better performance inversely proportional with the value of the score. This measure is to be taken with a (big) grain of salt because it does not distinguish between low values due to a perfect text recognition and low values due to a lack of matched nodes. The measure indicates the effort to correct the recognized labels, but this effort is also zero if the nodes are not recognized at all.

Figure 5 indicates that the MCS measure was able to clearly distinguish about half of the pairs of runs, which is relatively good considering how conservative the test is, compared to existing pair-wise tests such as the bootstrap test [18]. In terms of absolute difference between the runs, MCS shows significant difference above the 0.1 mark (of its 0 to 1 scale). The MTM and MXM are less able to distinguish between runs, and this further supports the idea that they should only be used as additional information to the MCS.

5.2 Query sets

The complementary evaluation was based on the queries automatically generated from the ground truth graphs, as described in Section 4.2. The evaluation is performed topic-per-topic. A set of queries for the entire collection was also created, but only a handful were found to actually return more than one topic image. (e.g. *"start"—"stop"*). We therefore continued to work on a topic-basis to keep in line with the graph-based measures. Figure 6 shows the number of queries per graph, ordered by the number of nodes in each graph. As expected, the set of queries grows exponentially with the number of nodes in the graph. For each query, we attempted to find a path in the run graph. The *dijk* method from Section 4.2 implements a slightly modified version of the Dijkstra method. Namely, it takes as parameters two node labels rather than two nodes, and returns the smallest path between any two nodes such that the starting node has the first given label, and the ending node has the second given label.

With this adaptation, we compute the recall and precision as defined above and plot the average recall (AR) and

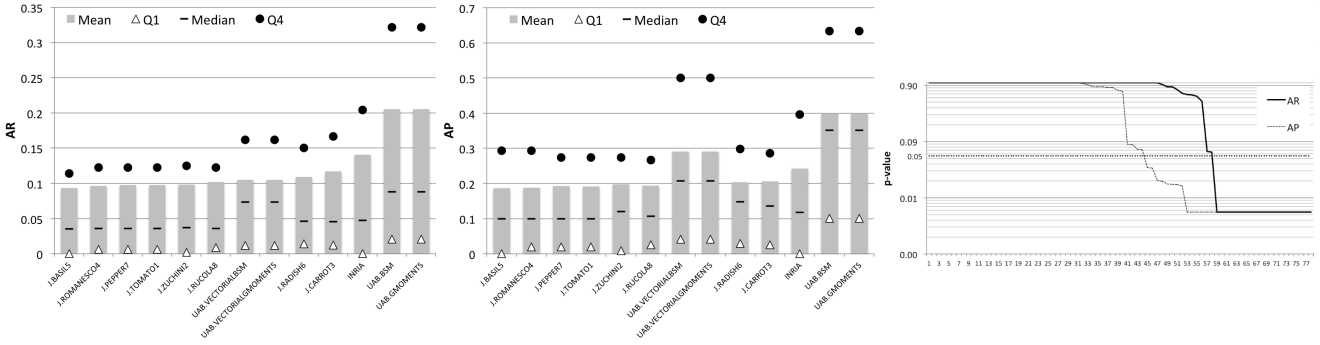


Figure 7: Query measures.

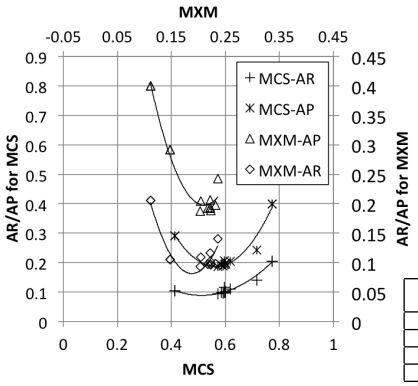


Figure 8: Correlations between values of graph and query-based measures

	Kendall τ
MCS-AR	0.533*
MCS-AP	0.384
MXM-AR	-0.24
MXM-AP	-0.384

Table 1: Rank Correlations: Graph - Query measures

average precision (AP) across all topics in Figure 7. Also in Figure 7, we plot the distribution of p-values for these two measures. The AR only manages to distinguish 26% of the pairs of runs, while the AP 44%. The lowest significant difference between two runs is for the AR 0.09. Looking again at Figure 7 we can see that such a difference appears only between the two top performing systems and the rest.

5.3 Correlations

With respect to how the two sets of measures correlate with each other in terms of system rankings, the expectation is that the MCS and MXM should correlate with AR, since whether a system recognized the structure and labels has a direct impact on its ability to answer the kind of queries we have defined. The relation between MCS/MXM and average precision (AP) is more difficult to estimate. We do not look at MTM-AR/AP at all since the node types play no role in the query generation. Table 1 shows the Kendall τ values for rank correlation for the four pairs. Correspondingly, Figure 8 plots the value correlations. We can see that the only statistically significant correlation is that between MCS and AR (but even this, only at the 0.05 level). In particular, we note the lack of correlation between the MXM and the AR.

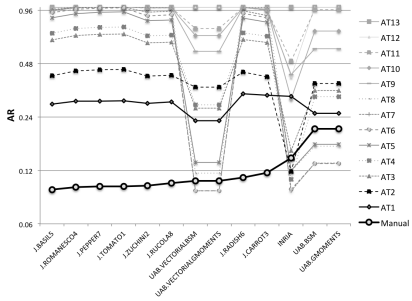
6. AUTOMATIC EVALUATION

The manual creation of ground truth for this task is a long and error prone task. For the 99 topics we started with, the average time to create the text representation of the flowchart was approximately 15 minutes. The effort is not only in duration, but also in the amount of care that the creator has to put in, to make sure that there are no missing or erroneous nodes and edges. Arguably, the task could be assisted by some specially-developed tool, but such a tool was not available and it is not clear whether one will be available.

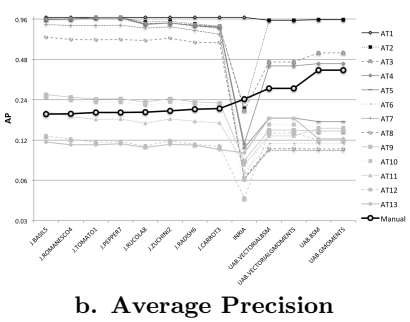
Therefore, we also considered the possibility of automated evaluation, following a simple voting mechanism, inspired by the Soboroff original work on the matter [21]. Such a method would apply particularly easily to the query-based evaluation. For the graph-based metrics, it is difficult to pinpoint which nodes correspond to which nodes, since even if we were able of identifying a maximal common subgraph of all the runs, it is not at all obvious how this MCS can be used in combination with the remaining nodes in each graph to determine what is the most likely correct solution.

We focus then on using the queries that we can generate from each run and for each topic to create a test set to apply back on each run. For the purposes of this initial test, we take a very simple approach, of pooling together all queries which are generated by at least n runs, with n varying between 1 and 13, the total number of runs in the collection. We thus create 13 sets of automatic truths (AT). The first and the last are particular cases. Namely, set AT1 contains all queries of all runs (7863 queries). Set AT13 we call the *veto* set because each run vetoes from addition to this set any query which it itself did not generate. Obviously, AT1 is expected to produce perfect values for AP for all runs, since all runs contribute all their results. Equally, AT13 cannot be used to estimate recall, since it will be perfect for each run. Figure 9 shows the results of this experiment. The AR and AP are calculated for each set and their results displayed in Figures 9a and 9b, respectively. We plotted the results on a logarithmic scale to make more visible the difference between the lower values, where the original ground truth results appear (indicated by the *manual* line on the plots). Figure 9c shows the number of queries in each set, as well as the number of topics which contributed to the query set. It also shows the Kendall τ rank correlation values between the manual and the different automatic results.

We can see that the influence of the different groups of runs are weighting heavily on the final ranking in all automatic truth sets and distort the final results to the point that there is either no correlation between the manual and automatic rankings, or the lower performing runs are dom-



a. Average Recall



b. Average Precision

Query Set	queries	topics	Kendall τ	
			AR	AP
AT1 (all)	7863	99	-0.05	-0.58*
AT2	4794	99	-0.46*	-0.21
AT3	2790	99	-0.46*	-0.63**
AT4	2577	99	-0.49*	-0.63**
AT5	2034	99	-0.52*	-0.71**
AT6	1883	98	-0.46*	-0.76†
AT7	1661	98	-0.52*	-0.79†
AT8	1232	95	-0.33	-0.76†
AT9	246	69	-0.44*	-0.76†
AT10	196	62	-0.46*	-0.76†
AT11	109	47	-0.34	-0.63**
AT12	59	37	-0.06	-0.03
AT13 (veto)	25	15	NA	0.119

c. Sets stats, Rank Correlation

Figure 9: Automatic evaluation results show little connection with manual results. The table shows the rank correlations between them with statistical significance indicated at 0.05 (*), 0.005 (**) and 0.0005 (†) levels

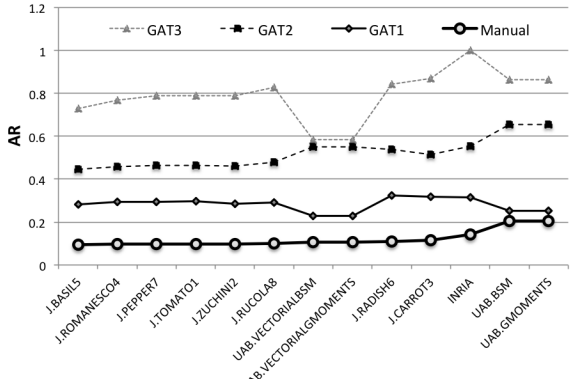


Figure 10: AR Results using only one vote per group

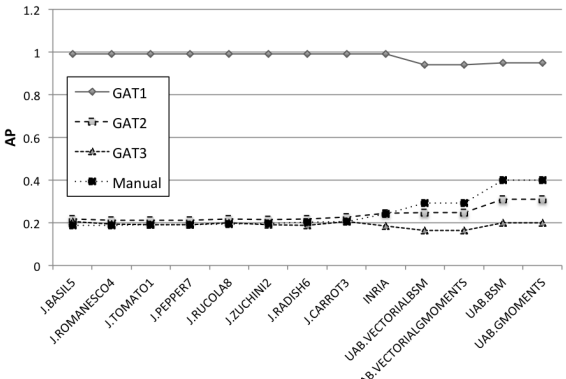


Figure 11: AP Results using only one vote per group

inating the rest and reversing the order (particularly so for AP). The INRIA run is particularly affected in this scenario, since it has no support from other runs within the existing collection. We can see in Figure 9b that as soon as two runs have to agree on a query to add it to the test collection (i.e. starting with AT2), the result of the INRIA run drops drastically.

Such results are in fact not unexpected, yet useful to understand the task at hand. The immediate corrective action to mitigate the heavy influence of a set of similar runs is to consider only one vote per participating team. We can then build three sets of queries, let us call them *Group Automatic Truths* (GAT), where a query is present if it was generated by at least one run of at least 1, 2, or 3 participating groups. Figures 10 and 11 show the thus obtained results in terms of AR and AP, respectively. Table 2 shows the number of queries and topics in each GAT set, as well as the correlation between the results obtained automatically and the manual results.

In this case again, two of the three sets are special cases, where either precision or recall are expected to reach maximal values (GAT1 and GAT3, respectively). However, because the different runs of a group are not quite identical, perfect scores are not reached for all the runs. The exception is again the INRIA run, which, being alone in its group, has a true veto for GAT3. The remaining set, GAT2, estimates quite well the manual ranking of systems. Both in terms of AP and AR it is able to identify the top and worst performing runs, while reversing the order of some mid-performing runs.

Query Set	#queries	# topics	Kendall τ	
			AR	AP
GAT1 (all)	7863	99	-0.053	-0.580*
GAT2	608	84	0.813†	0.842†
GAT3(veto)	112	30	0.554*	-0.263

Table 2: Sets statistics and Rank Correlation between automatic and manual evaluations when using the different sets of Group Automatic Truths. Significance given at the 0.05 (*) and 0.0005 (†) levels

7. CONCLUSIONS

We have considered here the problem of evaluating flowchart recognition tools for the purposes of a patent searcher. The problem can be viewed from two perspectives, based on two related use-cases for such an application. First, the more general use-case of taking an existing, perhaps hand-drawn, flowchart, and digitizing it for further processing. Second, the search-specific use-case of identifying patents describing processes or methods that relate two distinct concepts. A set of simple measures have been considered for each of these use-cases. An analysis in terms of their ability to distinguish runs and to correlate with each other leads to the following conclusions:

- a measure based on the maximal common subgraph (MCS) is highly capable to distinguish between runs (over 50% of run pairs), but its computational complexity makes it unusable for graphs larger than 25-30 nodes, in the absence of good anchors between the ground truth and the run graphs
- two contributing measures based on the percentage of

matched node types and the edit distance between two nodes add to the understanding of the results, but are meaningless outside the context of the MCS measure above. Therefore, they suffer from the same complexity problems.

- two measures based on the hypothetical set of queries which a graph should be able to respond to are easy to calculate (if we ignore issues related to duplicate labels within a graph) and are easily relatable to the task of searching for innovation, but have less distinguishing power (only 26% and, respectively 44% of run pairs can be significantly distinguished)

- automatic ground truth creation appears to work via a majority voting mechanism but only if not each run gets a vote, but rather each group. In such a case, a statistically significant correlation higher than 0.8 (Kendall τ) is achieved between the ranks assigned by the manual evaluation and those of the automatic evaluation.

- the correlation between rankings using the graph-based and the query-based methods is either insignificant or only slightly positive (τ=0.53). However, since no human participants were involved, it is impossible to say at this time which of the two is better. While we do start with well documented assumptions about user needs, a study similar to [20] would be desirable as future work.

The observations and results described in this report suffer from the relative small set of participating groups. However, all data is available on the CLEF-IP website[3], for interested research groups to test their own systems.

Acknowledgments

The authors were partially supported by the European Commission through the PROMISE Network of Excellence (Grant no. 258191) and by the Austrian Research Promotion Agency through the IMPEx Project (Grant no. 825846).

8. REFERENCES

[1] A.-M. Awal, G. Feng, H. Mouchere, and C. Viard-Gaudin. First Experiments on a New Online Handwritten Flowchart Database. In *Proc. SPIE 7874, Document Recognition and Retrieval XVIII*, 2011.

[2] H. Bunke, P. Foggia, C. Guidobaldi, C. Sansone, and M. Vento. A Comparison of Algorithms for Maximum Common Subgraph on Randomly Connected Graphs. In T. Caelli, A. Amin, R. Duin, D. Ridder, and M. Kamel, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 2396 of *LNCS*. Springer, 2002.

[3] B. A. Carterette. Multiple Testing in Statistical Analysis of Systems-based Information Retrieval Experiments. *ACM Trans. Inf. Syst.*, 30(1), 2012.

[4] A. Fujii, M. Iwayama, and N. Kando. Overview of Patent Retrieval Task at NTCIR-4. In *Proc. of NTCIR-4*, 2004.

[5] A. Järvelin, G. Eriksson, P. Hansen, T. Tsikrika, A. G. S. de Herrera, M. Lupu, M. Gäde, V. Petras, S. Rietberger, M. Braschler, and R. Berendsen. Deliverable 2.2 Revised Specification of the Evaluation Tasks. Technical report, PROMISE Network of Excellence, 2012.

[6] A. Lemaitre, H. Mouchere, J. Camillerapp, and B. Couasnon. Interest of syntactic knowledge for on-line flowchart recognition. In Y.-B. Kwon and J.-M. Ogier, editors, *Graphics Recognition. New Trends and Challenges*, volume 7423 of *LNCS*. Springer, 2013.

[7] M. Lupu and A. Hanbury. Patent Retrieval. *Foundations and Trends in Information Retrieval*, 7(1), 2013.

[8] M. Lupu, Z. Jiashu, J. Huang, H. Gurulingappa, I. Filipov, and J. Tait. Overview of the TREC 2011 Chemical IR Track. In *Proc. of TREC*, 2011.

[9] M. Lupu, R. Mörzinger, T. Schleser, R. Schuster, F. Piroi, and A. Hanbury. Patent Images - a Glass encased Tool / Opening the case. In *Proc. of iKnow Conference*, 2012.

[10] J. J. McGregor. Backtrack Search Algorithms and the Maximal Common Subgraph Problem. *Software Practice and Experience*, 12, 1982.

[11] H. Miyao and R. Maruyama. On-Line Handwritten flowchart Recognition, Beautification and Editing System. In *Proc. of ICFHR*, 2012.

[12] R. Mörzinger, R. Schuster, A. Horti, and G. Thallinger. Visual Structure Analysis of Flow Charts in Patent Images. In *Working Notes of CLEF*, 2012.

[13] F. Piroi, M. Lupu, A. Hanbury, A. Sexton, W. Magdy, and I. Filippov. CLEF-IP 2012: Retrieval Experiments in the Intellectual Property Domain. In *Working Notes of CLEF*, 2012.

[14] F. Piroi, M. Lupu, A. Hanbury, and V. Zenz. Clef-ip 2011: Retrieval in the intellectual property domain. In *CLEF (Notebook Papers/Labs/Workshop)*, 2011.

[15] J. W. Raymond and P. Willett. Maximum Common Subgraph Isomorphism Algorithms for the Matching of Chemical Structures. *Journal of Computer-Aided Molecular Design*, 16, 2002.

[16] M. Rusinol, L.-P. de las Heras, J. Mas, O. R. Terrades, D. Karatzas, A. Dutta, G. Sanchez, and J. Llados. CVC-UAB's participation in the Flowchart Recognition Task of CLEF-IP 2012. In *Working Notes of CLEF*, 2012.

[17] T. Sakai. Evaluating evaluation metrics based on the bootstrap. In *Proc. of SIGIR*, 2006.

[18] T. Sakai. Evaluation with Informational and Navigational Intents. In *Proc. of WWW*, 2012.

[19] M. Sanderson. Test collection based evaluation of information retrieval systems. *Foundations and Trends in Information Retrieval*, 4(4), 2010.

[20] M. Sanderson, M. L. Paramita, P. Clough, and E. Kanoulas. Do User Preferences and Evaluation Measures Line Up? In *Proc. of SIGIR*, 2010.

[21] I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. In *Proc. of SIGIR*, 2001.

[22] A. Thean, J.-M. Deltorn, P. Lopez, and L. Romary. Textual Summarisation of Flowcharts in Patent Drawings for CLEF-IP 2012. In *Working Notes of CLEF*, 2012.

[3] http://www.ifs.tuwien.ac.at/ clef-ip/2013/flowcharts.shtml