Internal distribution code:

(A) [-] Publication in OJ
(B) [-] To Chairmen and Members
(C) [-] To Chairmen
(D) [X] No distribution

Datasheet for the decision of 5 June 2023

Case Number: T 1867/18 - 3.5.06

Application Number: 08870601.5

Publication Number: 2235621

IPC: G06F7/00, G06F17/30

Language of the proceedings: EN

Title of invention:
MANAGING AN ARCHIVE FOR APPROXIMATE STRING MATCHING

Applicant:
Ab Initio Technology LLC

Headword:
Approximate string matching/AB INITIO

Relevant legal provisions:
RPBA 2020 Art. 13(2)
EPC Art. 56

Keyword:
Amendment after summons - taken into account (yes)
Inventive step - (no)

Decisions cited:
G 0001/19, T 0641/00, T 1730/11, T 1742/12, T 0697/17

Beschwerdekammern

Boards of Appeal

Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0
Fax +49 (0)89 2399-4465

Case Number: T 1867/18 - 3.5.06

D E C I S I O N
of Technical Board of Appeal 3.5.06
of 5 June 2023

Appellant: (Applicant)	Ab Initio Technology LLC 201 Spring Street Lexington, MA 02421 (US)
Representative:	Lloyd, Patrick Alexander Desmond Reddie & Grose LLP The White Chapel Building 10 Whitechapel High Street London E1 8QS (GB)
Decision under appeal:	**Decision of the Examining Division of the European Patent Office posted on 19 February 2018 refusing European patent application No. 08870601.5 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman	M. Müller
Members:	M. Domingo Vecchioni
	B. Müller

Summary of Facts and Submissions

I. The appeal is against the decision of the examining division, dated 19 February 2018, to refuse European patent application No. 08870601.5.

II. The examining division refused the application on the basis that claim 1 according to the main request and the first and second auxiliary requests did not fulfil the requirement of inventive step, Article 56 EPC, starting from a notorious general-purpose computer system or, alternatively, from prior art document

D1: T. Bocek et al., Fast similarity search in large dictionaries, Technical Report No. ifi-2007.02, Department of Informatics, University of Zurich, April 2007. [XP002679634]

The decision also cites the following document but does not rely upon it in the reasons:

D4: O. Hassanzadeh et al., Accuracy of approximate string joins using grams, VLDB'07, 23-28 September 2007. [XP055032377] Retrieved from the Internet on 11 July 2012, URL: http://www.cs.toronto.edu/~oktie/papers/qdb07.pdf

III. Notice of appeal was filed on 13 April 2018, the appeal fee being paid on the same day. With the grounds of appeal, filed on 19 June 2018, the appellant requested that the decision of the examining division be set aside and that a patent be granted on the basis of the main request or, alternatively, one of the first to third auxiliary requests, all submitted with the statement of grounds. The main request, the first and the second auxiliary requests were the same as those underlying the decision under appeal. Oral proceedings were conditionally requested.

IV. In an annex to a summons to oral proceedings, the board provided its preliminary opinion on the appeal. None of the requests appeared to meet the requirements of Articles 84 or 123(2) EPC. Also, the method according to claim 1 of all requests appeared to lack an inventive step, Article 56 EPC, in view of either a notorious general-purpose computer system alone, or in view of D1 and common general knowledge.

V. With a letter received on 16 November 2022 (hereinafter: "the reply to the summons"), the appellant filed claims for a new main request to replace those of all pending requests, conditional on the admittance of the new claims.

The appellant argued that the amendments made in the main request had been "made in good faith in direct response to objections [regarding clarity and added subject-matter] which ha[d] been newly raised in the Board's opinion accompanying the summons". These new objections represented "exceptional circumstances" within the meaning of Article 13(2) RPBA and therefore the new main request should be admitted.

Arguments in favour of the new main request as regards Articles 84, 123(2) and 56 EPC were also submitted.

VI. On 12 December 2022, the appellant indicated that it would not attend the oral proceedings and requested a decision based on the submissions filed with its letter of 16 November 2022.

VII. The oral proceedings were thereupon cancelled.

VIII. Independent claim 1 according to the main request reads as follows:

"a computer-implemented method for managing an archive for determining approximate matches associated with strings occurring in data records of a dataset, the method including:

pre-processing, by a pre-execution module (110), data records to determine a set of string representations that correspond to strings occurring in the data records;

generating, for each of at least some of the string representations in the set, a plurality of close representations that are each generated from at least some of the same characters in the string, the close representations comprising deletion variants of the corresponding strings;

calculating a frequency of occurrence in the data records for each of the at least some of the strings represented in the set of string representations;

comparing generated close representations of a first string to generated close representations of a second string, and identifying whether any of the close representations of the first string correspond to any of the close representations of the second string such that the first and second string are a potential approximate match;

storing entries in an archive that each represent a potential approximate match between at least two strings based on their respective close representations;

renormalizing the frequency of at least one string by summing the counts of the strings that are potential approximate matches of the at least one string and, based on the renormalizing, generating a significance value for the one or more strings that can be used for identifying further potential approximate matches, the generated significance value for the at least one string being stored in association with the at least one string; and

executing, by an execution module (112), a computation graph wherein a component of the computation graph accesses the archive to determine whether given data records should be processed based on whether strings in the given data records are a potential approximate match, and wherein the component of the computation graph consolidates the given data records having strings that are a potential approximate match."

Reasons for the Decision

The application

1. The application relates to approximate string matching (also called fuzzy string matching or searching), i.e. finding strings that match a given pattern string within some tolerance according to a similarity metric, such as the edit distance. The strings being searched

may be strings contained in records of a database (page 1 of the original description, lines 4-15).

2. Approximate string matching may be used in database operations like join or rollup that group ("consolidate") records into sets based on matching keys, in order to take into account that the exact spelling of words may differ within a dataset or between data sources and that words may be misspelled, e.g. COMPNY instead of COMPANY (page 4, lines 15-23, and page 8, line 15 to page 9, line 10).

3. Database operations like join or rollup may be expressed as and executed based on a "computation graph", i.e. in terms of a graph-based computation (page 5, lines 21 to page 6, line 2, page 7, lines 11-20, figure 2 and page 10, lines 13-16).

4. To increase the efficiency of join, rollup and other database operations that use approximate string matching, the application proposes, in a first phase (pre-processing phase), to pre-process the data records so as to determine "potential approximate matches" between strings in the data records and to store the results in an "archive". The archive is used in a second phase (run-time phase) to determine approximate matches when performing the database operations (page 4, lines 24-29, page 7, line 21 to page 8, line 3, page 9, lines 11-13, page 15, lines 18-29, and page 19, lines 17-22).

5. In the pre-processing phase, strings from the data records are collected in a "dictionary" (in which they are stored according to some "string representation", which may be the string itself), variants of them ("close representations") are generated by deleting

characters (up to a fixed number of deletions), and potential approximate matches between two strings are determined by comparing their respective sets of variants (page 7, line 21 to page 8, line 3, page 9, line 27 to page 10, line 12, page 15, lines 13-17, and page 19, line 28 to page 22, line 30).

6. Moreover, a "fuzzy match score" indicating the quality of each potential approximate match is computed and stored in association with the potential approximate match in the archive. In the run-time phase, a "potential approximate match" between two strings is identified as an actual "approximate match" based on the associated "fuzzy match score" (page 8, lines 1-3, page 16, lines 15-17, and page 21, lines 5-16).

7. Furthermore, during the pre-processing phase, the frequency of occurrence of strings in the data records may be calculated and "renormalized" by adding to it the frequency of variants that are potential approximate matches. For example, determining the frequency of the string COMPANY in the data records by counting also the number of times variants such as COMPNY occur in the data records (page 25, lines 5-11, and page 28, line 28 to page 29, line 15).

The renormalized frequencies may be used to compute a "significance score" for each string. This score may be used to identify likely misspellings or likely false positives, e.g. CLARKE and CLAIRE, or to match phrases (page 14, line 23 to page 15, line 7, and page 25, lines 5-25).

Admittance

8. The board agrees with the appellant's submission that the main request filed with the reply to the summons represents a good faith reaction to the objections under Articles 84 and 123(2) EPC which had been raised for the first time by the board in its communication pursuant to Article 15(1) RPBA accompanying the summons, on the basis of claims which were the same as those on which the decision under appeal was based, and accepts these circumstances as "exceptional circumstances" within the meaning of Article 13(2) RPBA. Taking also into account that the main request does not substantially alter the matter for which inventive step is to be assessed, the board exercises its discretion under Article 13(2) RPBA in admitting the main request into the proceedings.

9. Accordingly, the appellant is understood to have withdrawn the previous main request and first to third auxiliary requests (see point V above).

Inventive step

10. Only features of a claimed invention that contribute to its technical character may support the presence of an inventive step within the meaning of Article 56 EPC (T 641/00-*Two identities/COMVIK*, headnote 1; G 1/19-*Pedestrian simulation*, reasons 31 and 37-39).

11. The examining division found that the subject-matter of claim 1 of the then pending requests lacked an inventive step in view of two alternative lines of argumentations: a first one starting from a notorious general-purpose computer system, over which the claimed method did not make any non-obvious *technical* contribution,

and a second one starting from the method disclosed in prior art document D1.

12. In its preliminary opinion, the board endorsed essentially both lines of argumentations.

As regards the appellant's argument that neither starting point considered by the examining division was a suitable one, the board noted that the mere fact that a piece of prior art has a purpose different from that of the invention did not prohibit the consideration of an inventive step assessment starting from it (see e.g. T 1742/12-*On demand instantiation/RAYTHEON*, reasons 9). As a matter of principle, this also applied to the general-purpose computer.

It was furthermore permissible to start the assessment of inventive step by identifying which part of the claimed subject-matter contributes to its technical character, as only that part might support the presence of an inventive step within the meaning of Article 56 EPC.

13. The appellant argued in its reply to the summons that the claimed method achieved various technical effects: data records were updated, which was technical in view of T 697/17-*SQL extensions/MICROSOFT TECHNOLOGY LICENSING*, reasons 5.3.4; the output (consolidated) dataset was more accurate; the speed of execution of the computation graph was increased due to the pre-processing of the data records as specified in the claim, which was technical in view of T 1730/11-*Graph-based computation/AB INITIO*, reasons 4.9. According to the appellant, the objective technical problem solved by the method of claim 1 over D1 was "how to provide a computer-implemented method for managing an archive

that provides a more accurate dataset and an increase in the processing speed of data records".

14. The amendments made to claim 1 with the reply to the summons have specified in some more detail what is stored in the "archive" during the pre-processing phase and how it is used during the run-time phase to enable an efficient implementation of consolidation operations on data records (e.g. join or roll operations) in a dataset based on an approximate matching of strings occurring in these records.

In view of the function of the generated archive as a data structure used to enable an efficient implementation of database operations, the board tends to recognise a technical contribution in at least this aspect of the subject-matter of claim 1.

15. Therefore, the board considers it to be more adequate, if only from a pragmatic point of view, to assess inventive step for present claim 1 first in view of document D1, in which a similar data structure is used for a similar purpose.

For claim features that are either disclosed in D1 or obvious in view of the prior art and common general knowledge, there may then be no need to decide whether - and, if so, to which extent - they contribute to the technical character of the claimed invention, which simplifies the assessment task.

16. *Document D1*

16.1 D1 discloses an algorithm called "Fast Similarity Search (FastSS)" to search a query string in a dictionary of strings based on the edit distance as similarity function (D1: abstract; section 1, paragraphs 1 and 2).

D1 thus discloses an algorithm for "approximate string matching" in the sense of the present application.

The dictionary in D1 corresponds to the "dictionary (111)" in the present application.

16.2 D1 discloses further that the FastSS algorithm may be used in various application contexts (see abstract). Exemplary applications are finding similar words in a book (Moby Dick) or in Wikipedia articles. In these applications, the book chapters resp. the articles are stored in an SQLite resp. MySQL database (sections 5.1 and 5.2).

Hence, D1 discloses "determining approximate matches associated with strings occurring in data records of a dataset" as in claim 1.

16.3 The FastSS algorithm is an offline algorithm, i.e. an algorithm that "pre-process[es] the target data and [...] store[s] it in memory or on disk to speed up query processing" (section 2.7, paragraphs 1 and 2).

16.4 The pre-processing phase in D1 is explained in particular in section 3.2.1 "Indexing": For all words (i.e. strings) in the dictionary, and a given number of edit operations k, FastSS generates all variant spellings recursively and stores them in an "index" as tuples (v, x), where v is a dictionary word and x a list of deletion positions. The variant spellings are obtained by deleting up to k characters from the word. The set of variant spellings generated for a word v is called its "k-deletion neighborhood $U_d(v,k)$" (sections

3.2.1, 3.2.2 and 3.3).

The k-deletion neighborhood of a word/string in D1 corresponds to the "deletion set" for that string as defined in the present application from page 20, line 1 to page 21, line 1, and, in claim 1, to the "plurality of close representations" generated for a "string representation" (which in D1 is, for a given string, the string itself). They are thus "deletion variants of the corresponding strings" as in claim 1.

16.5 The run-time stage in D1 is described in particular in section 3.2.2 "Retrieval" and section 3.3. For a query word p, its k-deletion neighborhood is generated. Each variant in that neighborhood is looked up in the index storing all the variants of the dictionary strings and the associated lists of deletion positions. If a match between variants is found, the edit distance between the query string and the corresponding dictionary string can be derived from the respective lists of deletion positions (using the formula of theorem 4). If the edit distance is not greater than threshold k, the dictionary string is considered to be an approximate match for the query string.

Hence, like in the present application (see page 14, lines 9-22, and page 21, lines 5-16), the determination of an "approximate match" between two strings involves, first, identifying a "potential approximate match" between the two strings based on a comparison of their respective deletion sets (sets of close representations) and, secondly, computing a "fuzzy match score" for the quality of the match (in D1 the edit distance between the two strings based on the lists of deletion positions) and comparing it with a threshold. The first of these two steps carried out at run-time in D1 amounts, in the terms of claim 1, to a step of "comparing generated close representations of a first string to generated close representations of a second string, and identifying whether any of the close representations of the first string correspond to any of the close representations of the second string such that the first and second strings are a potential approximate match", with the first string being a query string and the second string a string occurring in the data records (the dictionary).

16.6 However, the method of D1 does not involve any *pre-computation* of potential approximate matches *between strings occurring in the data records*. In D1, the index stores the k-deletion neighborhoods (i.e. the sets of close representations) of the strings in the dictionary.

In the invention according to claim 1, the archive stores instead *potential approximate matches between such strings*.

These differences may be labelled **difference (1)**.

16.7 D1 does also not disclose a *calculation of frequencies of occurrence of strings*, a *renormalization of such frequencies* and a *generation of a "significance value" for at least one string based on the renormalization* and its *storage in the archive in association with the string*, as recited in claim 1.

These differences may be labelled **difference (2)**.

16.8 D1 does also not disclose *"executing, by an execution module (112), a computation graph wherein a component of the computation graph accesses the archive to* *determine whether given data records should be processed based on whether strings in the given data records are a potential approximate match, and wherein the component of the computation graph consolidates the given data records having strings that are a potential approximate match*", as recited in claim 1.

These differences may be labelled **difference (3)**.

16.9 The method of claim 1 thus differs from the method disclosed in D1 in differences (1) to (3).

16.10 In the reply to the summons, the appellant identified essentially the same differences between claim 1 and D1 ("differences (i) and (iii)" in that letter are included here in difference (1), "differences (ii) and (iv)" in difference (2), and "difference (v)" is difference (3)).

16.11 With respect to the feature "identifying whether any of the close representations of the first string corres- pond to any of the close representations of the second string such that the first and second strings are a potential approximate match", the appellant argued that this feature was entirely absent in D1 as D1 identified approximate matches on the basis of the edit distance.

The board does not follow this argument. As explained at points 16.5 and 16.6 above, such an identification step is carried out in D1, as a first step towards the identification of an approximate match, however at run- time - not in the pre-processing phase as in claim 1 - and for a query string and a string in the dictionary - not for two strings occurring in the dictionary (the data records) as in claim 1. These aspects in which claim 1 differs from D1 are included in difference (1).

17. *Obviousness of differences (1) and (3)*

The board considers that differences (1) and (3) would have been obvious to a skilled person starting from D1 in view of common general knowledge.

17.1 D1 discloses the use of the FastSS algorithm for finding words similar to a query string.

It is known to a skilled person that approximate string matching has further uses such as in the context of approximate join operations (see D4: section 1, paragraphs 1 and 2, and section 2, paragraph 1). It would thus have been obvious to consider how the FastSS algorithm disclosed in D1 could be applied to efficiently implement an approximate join operation.

17.2 The skilled person knows that pre-computation always requires a trade-off to be made between storage requirements and computation speed at run-time based on an identification of which calculations are expected to be frequently required at run-time.

An approximate join operation involves the merging of the data records of two datasets based on some key field. The required approximate string matching calculations concern exclusively pairs of strings occurring in these records (unlike the query search application primarily considered in D1, where the query string is unknown before run-time).

Hence, it would have been obvious to the skilled person that in such an application context, the pre-processing phase may go further and include not only the generation of the k-deletion neighborhood of all strings occurring in the datasets but also their potential approximate matches determined on the basis of the generated k-deletion neighborhood. This results in difference (1).

17.3 An approximate join operation is an operation that "consolidates" data records having strings in key fields that are an approximate match.

In the implementation at which the skilled person would have arrived starting from D1, as explained in the preceding point, the determination of whether two strings occurring in key fields of the data records are an approximate match would be made by looking up in the archive whether they are a potential approximate match and, if so, based on their edit distance.

This results in difference (3), except for the feature contained therein that the consolidation operation is realised as a "component of [a] computation graph".

17.4 However, whether the approximate join operation is to be executed as part of graph-based computations or not is, at least in the context of claim 1, a technically arbitrary choice. No aspect of the approach to approximate string matching used in the method of claim 1 is specifically adapted to be used in the context of graph-based computations, nor has this been argued by the appellant.

17.5 Hence, starting from D1, the skilled person would have arrived to differences (1) and (3) without any inventive activity. It may thus be left open to which extent they contribute to the technical character of the claimed invention.

17.6 The appellant argued in the reply to the summons that there was no teaching or suggestion in D1 of difference (1) "as D1 expressly adopts the edit distance model of string similarity over which the claimed deletion-join approach is an improvement".

The board is not convinced by this argument.

The present application presents the proposed approach as being faster than a basic approach that relies on computing the edit distance *for each pair of strings* to determine whether they are an approximate match. By first determining whether the strings are a potential approximate match, the "fuzzy match score" need only be computed for pairs of strings which are potential approximate matches, i.e. only for "close words". See page 14, lines 1-22, and page 19, line 28 to page 20, line 1 ("[r]ather than [to] compute a full edit distance between each pair of words, which would be expensive computationally, only nearby words are compared in the deletion-join procedure"). The computation of the "fuzzy match score" for two strings described on page 21, lines 5-16, amounts essentially to the computation of an edit distance for the two strings.

In D1 too, the edit distance is only computed for pairs of strings that are potential approximate matches (see section 3.3, first paragraph: "for each candidate") and the computation of their edit distance is performed in a very similar way to that of the "fuzzy match score" in the present application: see D1, section 3.3, first paragraph: "FastED implements Theorem 4, using deletion lists p1 and p2", with Theorem 4 describing a procedure very similar to that described on page 21, lines 5-18, in the present application.

Hence, the method disclosed in D1 is not to be equated with the basic "edit distance" approach described in the present application.

18. *No technical contribution by difference (2)*

The steps of calculating the frequency of occurrence of a string in the data records, renormalizing the frequency by taking into account the potential approximate matches of the string, generating a "significance value" for the string from the renormalized frequency and storing this value in the archive in association with the string - as specified in difference (2) - make no technical contribution to the method of claim 1 (beyond their implicit, not further defined computer-implementation).

18.1 Claim 1 is silent as to what is actually measured by the "significance value" generated for a given string. This can also not be derived from claim 1 as claim 1 does not specify how the significance value is generated from the renormalized frequency.

In the description, where this value is called "significance score", it is described as representing the inverse of the renormalized frequency of the string and thus "the relative importance of a word [i.e. string] to a phrase containing the word for the purpose of phrase comparison" (see page 11, lines 15-20, and page 28, lines 21 to 27).

The strings occurring in data records are abstract data, with no technical character. Determining by mathematical calculations their frequency, simple or renormalized, and their "significance" in the above sense is thus also - at least in itself - not technical.

18.2 The generated significance value also does not contribute to producing a technical effect in the context of the method of claim 1.

18.2.1 It is not derivable from claim 1 that the generated significance value is actually used in the context of the claimed method.

Claim 1 specifies, in the step of generating the significance value, that that value "*can* be used for identifying further potential approximate matches" (emphasis by the board) but claim 1 does not include any step in which it *is actually* used for that or any other purpose in the context of the claimed method.

The final step of the method of claim 1 specifies that a component of a computation graph "accesses the archive to determine whether given data records should be processed based on whether strings in the given data records are a potential approximate match" and that it "consolidates the given data records having strings that are a potential approximate match". This wording does not clearly require the significance value stored in the archive to be used in the consolidation operation. It could well be that only the potential approximate matches stored in the archive are used for that purpose, as only they are explicitly mentioned in relation to the consolidation operation.

18.2.2 It is also not apparent from the description how the significance value could be used for identifying *further* potential approximate matches, i.e. potential approximate matches not identified in the preceding step of "comparing generated close representations [...] and identifying whether any of the close representations [...] are a potential approximate match".

The described uses of the "significance score" (as the significance value is named in the description) appear to be confined to the identification of "false positives" when matching phrases or records, i.e. that a potential approximate match *identified in the preceding step* is not to be considered an actual approximate match. This is in particular the case in all the passages cited by the appellant as basis for the feature concerning the significance value, i.e. page 8, lines 7 to 9, page 11, lines 16 to 20, page 15, lines 2 to 5, and original claim 11.

The board notes that the significance value or score is distinct from the "fuzzy match score" (see page 11, lines 11-20).

18.2.3 It follows that the step of storing the significance value in association to the corresponding string - as specified in difference (2) - does also not make any technical contribution (beyond the implicit, not further defined computer-implementation of that step).

18.2.4 The appellant argued in the reply to the summons in respect of difference (2) that the significance value contributes to a technical effect in that it helps to deal with false positives and thus to ensure that "a more accurate output dataset is achieved".

The board is not convinced by this argument, if alone because it cannot be derived from the claim that the significance value is used to determine the output (consolidated) dataset.

Anyway, using a significance value to identify false positives in potential approximate matches is not *by itself* a technical use given the abstract nature of approximate string matching. Hence, even if this potential use were considered to be implied by claim 1 (it is not), it would not be an implied *technical* use in the sense of G 1/19.

18.2.5 As to the other alleged technical effects put forward by the appellant (see point 13 above), in particular increased computation speed, they have not been specifically linked to difference (2) but to differences (1) and (3), and they cannot anyway be relied on for difference (2) as the significance value is not used in the context of the method of claim 1.

18.3 Difference (2) does thus not contribute to the technical character of the method of claim 1 (beyond its implicit, not further defined and thus obvious computer-implementation). Consequently, it cannot support the presence of an inventive step.

19. *Conclusion on inventive step*

It follows that the method of claim 1 does not involve an inventive step within the meaning of Article 56 EPC over D1 and common general knowledge.

Concluding remarks

20. As the only request on file is not allowable, the appeal is to be dismissed.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar: The Chairman:

L. Stridde Martin Müller

Decision electronically authenticated