Technical Documentation

The Rise of Process Claims: Evidence from a Century of U.S. Patents*

Bernhard Ganglmair† W. Keith Robinson‡ Michael Seeligson§

May 4, 2021

Abstract

Patent claims define and describe the protected aspects of an invention. Innovation researchers have taken advantage of the recent digitization of patents to conduct large-scale analysis on the text of the claims. While much work has been applied to describing the industry or technology, our analysis focuses on the form of the innovation covered in patents. We present our methodological approach and descriptive results from a systemic classification of U.S. patent claims, the "**PAT**ent **C**laim **C**lassification by **A**lgorithmic **T**ext-Analysis" (`patccat`). Using the texts of granted U.S. utility patents, we classify all independent claims along dimensions of claim class (process vs. product) and type (e.g., product-by-process claims, means-plus-function claims, Jepson claims). This data appendix to Ganglmair, Robinson, and Seeligson (2021) provides a detailed account of `patccat` and the data we obtain from it.

Keywords: cats; patent claims; text analysis; patents.

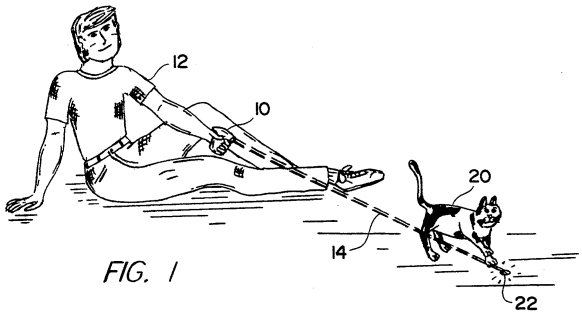
JEL Codes: C81; O31; O34; Y10

*We thank Florence Blandinieres, Thomas Hiesberger, Imke Reimers, and participants at PatCon7 (Chicago) and IPSDM (Mexico City) for helpful comments and suggestions. We also thank Jacob Colling, Maximilian Schneider, Lion Szlagowski, Jake Walsh, and Tianxiang Zhang for research assistance.

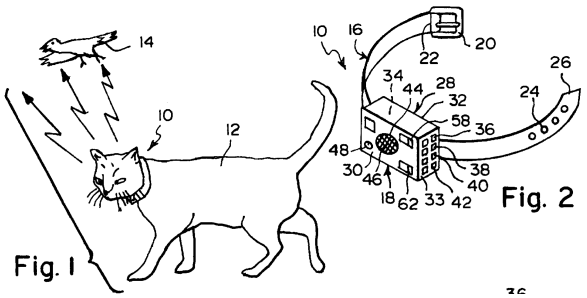
†University of Mannheim and ZEW Mannheim, Germany. E-mail: b.ganglmair@gmail.com.

‡Southern Methodist University, Dedman School of Law, Dallas, TX, USA. E-mail: wrobinson@smu.edu.

§University of Texas at Dallas, Naveen Jindal School of Management, Richardson, TX, USA. E-mail: seeligson@gmail.com.



U.S. Patent 5,443,036
("Method of exercising a cat")



U.S. Patent 5,952,925
("Collar for a cat for warning a bird of the presence of the cat")

1 Introduction

The acronym `patccat` stands for "**PAT**ent **C**laim **C**lassification by **A**lgorithmic **T**ext-Analysis". This data appendix to Ganglmair, Robinson, and Seeligson (2021) provides a detailed account of `patccat` and the data we obtain from it. The current version is Release 3 (December 2020).

The main classification is along the line of the invention type. Each independent claim is classified as a process claim, product claim, or product-by-process claim. In addition, we provide information on whether a claim is a Jepson claim or a means-plus-function claim. The data and documentation are available at

https://sites.google.com/site/bganglmair/data#patccat

The data files contain patent-level information (see Table 7 below for a list of variables) for patents granted between 1920 and 2020 – the time window for the results presented in Ganglmair, Robinson, and Seeligson (2021). We further provide (in a separate file) the results from our classification for all patents granted between 1836 and 1919.[1]

The technical documentation is accompanied by a number of files (see Table 1) that will allow users to run our classifier on any claims text input file that follows minimal formatting requirements. We have also added sample files containing the texts and manual classifications of 1,000 claims.[2]

We make both code and data freely available for others to use. In return, we ask users to cite the paper:[3]

[1] Users interested in claim-level information (see Table 6 for a list of variables): please contact the authors for the data files.

[2] The performance assessment results in Ganglmair, Robinson, and Seeligson (2021) are based on a broader sample (close to 10,000 claims) of manually classified patent claims.

[3] Please reach out and let us know of your projects (and results). We will keep a list of projects and papers that use data/code. If you want your work to be featured on this list, please let us know.

Table 1: Files in the `patccat` Documentation

File name	Description
`01-patccat-functions.R`	R script with the functions used in `patccat`
`patccat-functions.rda`	R object with the generated functions
`02-patccat-parameters.R`	R script with the parameterization used for the data construction in Ganglmair, Robinson, and Seeligson (2021)
`patccat-parameters.rda`	R object with the parameter objects
`03-patccat-classifier.R`	R script with the basic workflow of patent claim classification
`sample1000-claims-multi.csv`	File with a sample of 1000 claims in multi-line formatting
`sample1000-claims-single.csv`	File with the same sample of 1000 claims in single-line formatting
`sample1000-cats.csv`	File with the classification of the sample of 1000 claims (for benchmarking)

Ganglmair, Bernhard, W. Keith Robinson, and Michael Seeligson (2021): "The Rise of Process Claims: Evidence from a Century of U.S. Patents," Unpublished manuscript, University of Mannheim.

2 Approach

For our classification of patent claims, we combine information obtained from both the preamble and the body of a claim. The preamble is a general description of the invention (e.g., a method, an apparatus, or a device), whereas the body identifies elements and steps (specifying in detail the invention laid out in the preamble) which the applicant is claiming as the invention.

2.1 The Preamble

The classification of the preamble is based on a simple keyword search. We identify four different preamble types: We use the letter labels in Table 2 and in the data files to identify different preamble-body combinations.

"M" – Process preamble has a process keyword listed before a product keyword (if any);
"P" – Product preamble typically has a product keyword listed before a process keyword (if any).
"B" – For the classification of product-by-process claims, we examine the preamble of a given claim for the use of a "by-process" phrase.
"E" – Empty (or null) preamble that does not fall into one of the three main categories.

We provide more details, including keyword lists (for process claims and product claims) further below.

2.2 The Body

We complement the preamble type with information on the type of the body of the claim. The body is the part of the claim that describes the individual components of the invention. For the classification of the body, we take a parts-of-speech approach, analyzing the linguistic structure of each (indented) line or bullet point in the body of the preamble. The steps of a method or process (in a process claim) are listed using the gerund form of a verb, whereas the elements of an apparatus or device (in a product claim) are listed using nouns. The classification of each line primarily depends on whether a noun occurs before a gerund or whether a gerund occurs before a noun. The keywords themselves could even be the same: "applying paint using a brush" should be interpreted as a step in a process claim, but "a brush for applying paint" should be interpreted as an element in a product claim. We identify four body types:

"m" – Process body if the predominant share of lines in the body constitute steps;
"p" – Product body if the predominant share of lines in the body constitute elements;
"x" – Mixed body if steps or elements (or both) can be identified, but neither rise to the level of predominant;
"e" – Empty body if none of the individual lines in the body can be identified as either step or elements.

2.3 Preamble + Body = Claim

In a final step of our approach, we combine the classifications of the preamble and the body to obtain the invention type of a given claim. Each preamble-body combination translates into a claim type. We summarize our classification rules – from a preamble-body combination to a claim type – in Table 2.

Our approach prioritizes the preamble type and uses the body type as tie-breaker or when the information obtained from the preamble is inconclusive. We take a conservative view of this "preamble-first" approach by keeping the list of process words (for process preambles) and product words (for product preambles) short. Both lists comprise statutory terms in addition to a few other terms. Short lists imply that many edge cases are classified as mixed preambles (M). For such mixed preambles, we rely on information from the body for claim classification.

We classify claims with process preambles as process claims for all types of bodies. This also applies to single-line claims for which a body cannot be identified (and M–). Likewise, claims with product preambles are product claims – with one exception. A claim with a product preamble (P) but a process body (m) is a product-by-process claim (Pm). Claims with by-process preambles (B) (that explicitly refer to a process or method by which something is made or implemented) are product-by-process claims except when the body is a product body (p). In this exception (Bp), the body does not describe the process as announced in the preamble, and we consider the preamble

Table 2: Classification Table

Preamble	Body	Claim	Label
Process	Process	Process	Mm
Process	Product	Process	Mp
Process	Mixed	Process	Mx
Process	Empty	Process	Me
Process	—	Process	M–
Product	Process	Product-by-Process	Pm
Product	Product	Product	Pp
Product	Mixed	Product	Px
Product	Empty	Product	Pe
Product	—	Product	P–
By-Process	Process	Product-by-Process	Bm
By-Process	Product	Product	Bp
By-Process	Mixed	Product-by-Process	Bx
By-Process	Empty	Product-by-Process	Be
By-Process	—	Product-by-Process	B–
Empty	Process	Process	Em
Empty	Product	Product	Ep
Empty	Mixed	No Claim Type	Ex
Empty	Empty	No Claim Type	Ee
Empty	—	No Claim Type	E–

misleading. Last, when the preamble is empty (E), then the body is the decisive factor. A claim with a process body is a process claim (Em), and a claim with a product body is a product claim (Ep). For a mixed body, an empty body, or no body (Ex, Ee, E–), we assume that not enough information is available to classify the claim.

In `02-patccat-parameters.R`, we specify these rules in the matrix `-my.claim.rules-`. Function `-fn.claimtype-` uses this matrix as input. Some of our rules are chosen to optimize the results for a benchmark sample. A different context may require changes to these rules. We invite researchers to adapt our classification table (in particular for edge cases) for better fit.

3 Data Sources

Our main data source are the USPTO's Patent Grant Full Text Data files available at https://bulkdata.uspto.gov (Bulk Data Storage System BDSS), PatentsView at https://patentsview.org/download/data-download-tables, and Google Patents Public Data.

For patent claims that were initially filed and published with indented lines, we are able to preserve this multi-line structure and utilize it in our body classification. For patents, where this multi-line structure does not apply or is no longer preserved (the claims obtained from the Google Patents Public Data and PatentsView), our classifier tries to convert the text of each claim from

Table 3: Data Sources

Source	Format and notes	Sample
USPTO	XML format	2002–2020
USPTO	fixed-width text (APS, Green Book)	1976–2001
USPTO	fixed-width text (PATFT, Red Book)	1971–1975
PatentsView	data tables (claims in single-line formatting only)	1976-2020
Google	BigQuery	1920–1970

Notes: In Ganglmair, Robinson, and Seeligson (2021), we present descriptive results for patents granted between 1920 and 2020. On the project website, we also provide data for patents granted between 1836 and 1919 (obtained from the Google Patents Public Data).

single-line format to multi-line format. The following claims from U.S. patents 5,443,036 and 5,952,925 are examples of claims in multi-line format.

We provide results for independent claims only – independent claims do not refer to other claims. Alternative data sources are, of course, possible. Our classifier can be applied to any claims text file with a unique patent-claim identifier and the (single-line or multi-line) text of the patent claim. If a patent number and a claim number are provided, the classifier function `-fn.patccat-` will first construct a patent-claim identifier. If the text is of multi-line format, an additional variable that indicates the level of indentation (where level 1 is the preamble and levels 2 or higher are for the body) can be included. If this variable is not included, then the first line for a given patent claim is taken to be the preamble, and all other lines are body lines at the same level of indentation (level 2).

4 Workflow

In this section, we describe our workflow for the classifier in detail. The function `-fn.patccat-` performs the classification. It calls on a number of other functions that perform individual steps of our classifier. Function `-fn.patccat-` takes as input an R object `data.frame -data-` with the claims text, the column name for the patent-claim identifier (`-varPatentClaim-`), the column name for the claim text (`-varText-`), the column name for the level of indentation (`-varLevel-`, not required), the column name for the sequence in which the individual lines of a claim appear in the patent (`-varSequence-`, not required), and the column name of a simple running index (`-varID-`, not required). If a patent-claim identifier does not exist in the data, but a patent identifier and a claim identifier do, the function takes the respective variables names as inputs (`-varPatent-` and `-varClaim-`). See Table 5 for more details.

Table 4: Patent Examples

U.S. Patent 5,443,036: Method of exercising a cat		
varPatentClaim	varText	varLevel
5443036-0001	1. A method of inducing aerobic exercise in an unrestrained cat comprising the steps of:	1
5443036-0001	(a) directing an intense coherent beam of invisible light produced by a hand-held laser apparatus to produce a bright highly-focused pattern of light at the intersection of the beam and an opaque surface, said pattern being of visual interest to a cat; and	2
5443036-0001	(b) selectively redirecting said beam out of the cat's immediate reach to induce said cat to run and chase said beam and pattern of light around an exercise area.	2
U.S. Patent 5,952,925: Collar for a cat for warning a bird of the presence of the cat		
varPatentClaim	varText	varLevel
5952925-0001	1. A collar for wearing by a cat and for warning a bird of the presence of the cat, comprising:	1
5952925-0001	a) a strap for wearing around the neck of the cat; said strap being elongated and slender, and being selectively maintained around the neck of the cat, by a buckle affixed to one free end of said strap cooperating with throughbores in the other free end of said strap; and	2
5952925-0001	b) an electric device disposed on said strap; said electric device comprising a box being hollow and attached to said strap, at a position generally midway between said one free end of said strap and said other free end of said strap for allowing said box of said electric device to be worn under the chin of the cat; said box of said electric device having a front wall, a rear wall being spaced behind said front wall of said box of said electric device and attached to said strap, and a side wall extending from said front wall of said box of said electric device to said rear wall of said box of said electric device, and containing an electronic circuit; said electronic circuit of said electric device comprising a first pair of dip switches being recessed in, which prevents the cat from accessing them, and accessible from, said side wall of said box of said electric device.	2

4.1 Reformat

Our keyword search approach for the preamble classifier requires that the signal terms in the list of process words and product words appear at the beginning of the preamble. Likewise for the body classifier that searches for nouns or gerund forms of the verb at the beginning of each line of the body. Most of the claims in our data come well formatted. For the rest, we perform three steps to convert the claim text into a format we can process. Function `-fn.reformatdata-` performs this conversion. The function takes a `data.frame` with the claims text as input. The function calls on three separate functions: `-fn.jepsonreformat-`, `-fn.singlesplitter-`, and `-fn.beginWithIn-`. We describe each below.

Table 5: Input Data (`data.frame -data-`)

Variable	Description	Format
	Required Inputs	
`varPatentClaim`	Unique identifier for a patent claims (as patent-claim combination)	integer, string
`varPatent`	*Patent number/identifier*	*integer, string*
`varClaim`	*Claim number/identifier*	*integer, string*
`varText`	The text of the given line of the claim, including the leading outline designation (1., a., A., etc.)	string
	Additional Inputs	
`varLevel`	Level of indentation of a line. For a multi-line formatted claim, the preamble as highest-order line of a claim has a value of `varLevel` = 1; its body lines that are indented once have a value of `varLevel` = 2. All body lines with further indentations have higher values for `varLevel`. For a single-line formatted claim, the single line has a value of `varLevel` = 1.	integer
`varSequence`	Ordered sequence number of lines within a `varPatent`. `varSequence` equal to 1 is the first line (preamble if multi-line claim) of the first claim of the patent. The first line of a claim (the preamble if multi-line claim or the entire claim if single-line claim) is the smallest value of `varSequence` for a given `varClaim`.	integer
`varID`	Unique line identifier. It is not directly used by the classifier, but serves as a useful identifier for each row.	integer ≥ 0

4.1.1 Jepson Claim Reformatting (`-fn.jepsonreformat-`)

The preamble of a Jepson (or improvement) claim first describes what is known (or in the prior) art ("prior-art part"), followed by a transitional phrase (such as "wherein the improvement comprises"). After this transitional phrase, the claim lists everything that is considered an improvement ("improvement part"). For our preamble classifier, we use only the improvement part. We therefore split the preamble at the transitional phrase and treat the text of the improvement part as the text of the preamble. We do not use the prior-art part for our analysis.

4.1.2 Single-Line Claim Splitting (`-fn.singlesplitter-`)

Claims that are in single-line format (where the preamble and all lines of the body are concatenated and appear as one line or paragraph), are not useful for our preamble-body approach. We convert such claims into multi-line claims before applying our classifier. For this, we take two steps: First, we split the claim at the transitional phrase (e.g., "comprising the steps of" in patent 5,443,036 or "comprising" in patent 5,952,925) or certain punctuation characters to obtain the preamble and the body. Second, we identify enumeration counters in the text of body and use these to split the body into individual lines. Converted single-line claims are of one of three types:

1. Fully converted claims have a preamble and a multi-line body. We treat them as proper multi-line claims and apply the baseline version of our classifier.

2. Partially converted claims have a preamble and a single-line (non-converted) body. We treat this single-line body as the only line in the body and apply the baseline version of the classifier.

3. Non-converted claims are single-line claims that cannot be converted. By definition, they have an empty body. We apply a single-line version of our classifier.

4.1.3 In-Environment Claims (`-fn.beginWithIn-`)

A third (and relatively small category) of patents for format conversion are those beginning with the word "in" or "for" and a statement of the environment. Such claims have the structure of Jepson claims, but do not explicitly specify an improvement (and lack the respective transitional phrase). We trim the beginning of the preamble up to the first comma and use the text following the first comma for our preamble classifier.[4]

4.2 Preamble Type

The preamble classifier uses two keyword lists. One list (process words) with terms identifying a process preamble, a second list (product words) with terms identifying a product preamble. We choose short lists with statutory terms and a few strong terms to prioritize the preamble information in our claim classifier. To identify the preamble as either a process preamble or product preamble, a word from the process list or product list must appear in the first 8 words of the preamble. For single-line claims (for which we have not been able to convert the text of the claim into its multi-line claim format), this means that the respective words appear in the first 8 words of the claim.[5]

Function `-fn.preambletype-` performs the preamble classification. It takes several items as inputs. The parameter values are specified in `-my.params-` in R script `02-patccat-parameters.R`.

- a `data.frame` with the claims text
- a list of process words (`-processwords-`)
- a list of product words (`-productwords-`)
- a number of parameter values used for the classification
- a `TRUE` or `FALSE` flag of whether the claims are single-line claims

[4]The output from this function is a list with two items: a `data.frame` (`[data]`) with the converted claims texts (same structure as input), and a data.frame (`[df]`) with `-PatentClaim-` and a flag for the claim type.

[5]Underlined numbers are parameters specified in the object `-my.params-` in R script `02-patccat-parameters.R`. We have chosen these values as they optimize the accuracy of our classifier (using a small benchmark sample). Different contexts may require different values. We invite researchers to change them as needed!

4.2.1 Process Preamble

A process preamble is the preamble of a process claim. It names a method or process as the invention described by the claim. The two main words of the list of process words are indeed "process" and "method" as the most widely used terms to describe a process claim (e.g., in patent 5,443,036). A preamble is a process preamble if, within the first 8 words of the preamble, one of the process words is used, but none of the following applies: (1) one of the product words is used before the process word, (2) the process word is immediately followed by a noun, or (3) the terms for or by are used within 3 words of the process word.

4.2.2 By-Process Preamble

A preamble is a by-process preamble if it uses a by-process phrase. Such a phrase is "by [up to 3 words] process" or "by [up to 3 words] method." The preamble is not a by-process preamble if the preamble is a process preamble. In single-line claims, we search for by-process phrases in the first 50 words of the claim.[6] In multi-line claims, we impose no such word limit but consider the text of the entire preamble.

4.2.3 Product Preamble

A product preamble is the preamble of a product claim. It names a machine, an apparatus, or a device (a "thing") as the invention described in the claim. For our product-word list, we use statutory terms and a short list of very common terms used to describe "things." A preamble is a product preamble if, within the first 8 words of the preamble, one of the product words is used, but none of the following applies: (1) one of the process words is used before the product word, or (2) the preamble uses a by-process phrase.

The lists of keywords for process preambles and product preambles are specified in R script `02-patccat-parameters.R`. The object `-my.process.words-` contains the process words, the objects `my.product.words.short-` and `-my.product.words.long-` contain product words. The two lists for product words differ in their length. Only one of the lists is used, and a variable `-my.product.words-` is used in `-fn.patccat-` that points at one of the two lists. The long product word list contains the 100 most frequently used product words in a sample of 1% of all claims for patents granted between 1976 and 2015 (this list includes the short list).[7]

The word collar in the first claim of patent 5,952,925 is in neither product-word list. The preamble of that claim is thus an empty preamble.

[6] We restrict the number of words for single-line claims to minimize the noise from the language of the body in such claims.

[7] The list of process words is "method", "process", "approach", "manner", "practice", "recipe", "scheme", "technique", and "treatment". We also consider a "computer-implemented method" or "computer-implemented process." The list of product words is "system", "apparatus", "device", "machine", "computer", "assembly", "circuit", "data", "semiconductor", "composition", "medium", and "means".

4.3 Body Type

The body of a claim contains a number of (indented) lines of text, where lines describe steps (of a method or process) or elements (of an apparatus, device, or machine). Our approach for the classification of lines is a parts-of-speech approach. We classify the body in two steps. First, we identify each line in the body as either a (1) step, (2) element, or (3) a line that refers to other parts of the claim via the terms "said," "wherein," "whereby," or similar. Second, if the lines of a body are predominantly steps, then the body is a process body; if the lines are predominantly elements, then the body is a product body.

The function `-fn.bodytype-` performs the body classification. It takes several items as inputs: a `data.frame` `-data-` and a number of parameter values used for the classification. The parameter values are specified in object `-my.params-` in R script `02-patccat-parameters.R`. The function `-fn.bodytype-` also calls on function `-fn.POStagger-` that performs the parts-of-speech tagging. We use the openNLP POS tagger in the openNLP library.

4.3.1 Steps

In most cases, a line is a step if the gerund form of a verb occurs within the first 2 words of the line (e.g., "directing" or "redirecting" in patent 5,443,036) and none of the following applies: (1) the gerund form of the verb is from a list of commonly used words that do not describe steps of a method or process;[8] (2) a noun is used before the gerund form; (3) the word "means" is used in the line; (4) the line begins with words "said," "when," "wherein," "whereas," or similar; or (5) the line begins with a cardinal number or a determiner.

4.3.2 Elements

A line is an element if a noun occurs within the first 10 words of the line (e.g., "strap" and "device" in patent 5,952,925) and none of the following applies: (1) the line is a step; or (2) the line begins with words "said," "when," "wherein," "whereas," or similar. A line is also an element if it uses the word "means" (indicating a means-plus-function claim) or if one of the following constructions are used: (1) a noun is sandwiched between a gerund form of a verb and another form of a verb; (2) a noun is immediately preceded by a pairing of a cardinal number or determiner and a gerund; or (3) a noun is immediately preceded by a triple of a cardinal number or determiner, an adjective, and either an adjective or a gerund.

[8] This list contains the terms "being", "comprising", "consisting", "including", "having", "depending", "indicating", "representing", "containing", and "housing".

4.3.3 Classify the Body

Natural language processors may occasionally misclassify words; context and sentence structure is important, and while patent attorneys deploy a more predictable set of linguistic patterns than would be found in works of literature, errors may occasionally arise. It is possible, even less common, that those errors will lead to a misclassification of a line as a step instead of an item, or vice versa. But the body classification considers all of the line classifications together, minimizing the determinative impact of such rare cases. For the classification of the body, we aggregate the information obtained for each line. We consider four different body types. (1) The body is a process body if 90% or more of all lines of a body (classified as either steps or elements) are steps. (2) The body is a product body if 90% or more of all lines of the body are elements. (3) The body is a mixed body if at least one line is either a step or an element, but neither line type dominates the body. (4) The body is empty if it has neither steps nor elements.

4.4 Classifying the Claim

4.4.1 Multi-line Claims

Each preamble-body combination corresponds to a claim type as summarized above in Table 2. In the last step, we apply our set of preamble-body rules and classify claims. If there is no information from the preamble (empty preamble) and no information from the body (empty body), then we assume there is insufficient information for claim classification. In this case, the claim comes without a claim type.

4.4.2 Single-line Claims

For single-line claims that cannot be converted into multi-line claims, the claim type follows the preamble type.

4.5 Simple Approach to Process and Product Claims

We compare our results from the preamble-body classification approach above with those from two simpler approaches where we take a keyword approach to identify process claims. All other claims are assumed product claims.

`processPreamble` – In the first of these approaches, a claim is a process claim if in the preamble either "process" or "method" are used; and a product claim otherwise. This approach requires separate processing of the preamble and the body of the claim.

`processSimple` – In the second approach, we relax this requirement and classify a claim as a process claim if either the term "process" or "method" are used anywhere in the claim; and a product claim otherwise.

Note that unlike our full preamble-body classification, these two approaches yield full coverage, meaning that all claims can be classified. The function `-fn.process.simple-` performs this classification. It takes as inputs a `data.frame` `data` and a list with process words (here: "method" and "process", found in object `-my.processwords.simple-` in `02-patccat-parameters.R`).

5 Additional Claim-Level Information

In addition to the invention-type classification, we construct a number of additional variables related to types and format of patent claims.

- Jepson Claims (in function `-fn.jepson-`): A claim is a Jepson claim if it uses the term "improvement" or, if the term improvement is not used but the preamble begins with the preposition "in," it uses a term beginning with the character string "improve" in the preamble or the first line of the body.

- Means-plus-function Claims (in function `-fn.means-`): A claim is a means-plus-function claim if it uses the term "means" either in the preamble or the body (as part of an element in product body) of a claim.

- Text Length (in function `-fn.textlength-`): We calculate the length of the claim, separately for preamble and body (if a body exists or can be identified).

6 Implementation

In R script `03-patccat-classifier.R`, we provide the program lines that implement our approach. It loads the necessary RDA files (produced by R scripts `01-patccat-functions.R` and `02-patccat-parameters.R`) and data files. The file performs 4 steps:

1. Load functions, parameters, and data
2. Perform classifier (function `-fn.patccat-`)
3. Function `-fn.benchmark-` summarizes accuracy and coverage of the output; the function `-fn.diagnostics-` provides additional diagnostics. Both functions use the attached sample of 1000 claims in `sample1000-claims-multi.csv` (or `sample1000-claims-single.csv` for single-line claims) and their manual classification results in `sample1000-cats.csv`.
4. Function `-fn.rule.testing-` allows for simple accuracy checks of alternative preamble-body rules.

7 Data in the patccat Database

The output of function `-fn.patccat-` is a `data.frame`. We provide the list of variables and their description in Table 6. In Table 7, we provide the list of variables for the patent-level information.

References

Ganglmair, B., W. K. Robinson, and M. Seeligson (2021): "The Rise of Process Claims: Evidence from a Century of U.S. Patents," Unpublished manuscript, University of Mannheim.

Table 6: Claim-Level Information

Variable Name	Description	Values
`PatentClaim`	Patent-claim identifier of the form [patent number]-[claim number]	string
`singleLine`	Is claim in the input data in single-line format?	0 = no; 1 = yes
`singleReformat`	Format of the claim after conversation of function `-fn.singlesplitter-`	0 = multi-line claim (original); 1 = multi-line claim (converted); 2 = two-line claim (preamble and single-line body); 3 = single-line claim (not converted)
`Jepson`	Is claim a Jepson claim?	0 = no; 1 = yes
`JepsonReformat`	Format of the claim after conversion of the function `-fn.jepsonreformat-`	0 = not Jepson claim; 1 = Jepson claim (converted)
`inBegin`	Does claim begin with an "in" phrase?	0 = no; 1 = yes
`lengthPreamble`	Length of the text of the preamble (in characters); length of claim if single-line format	integer
`lengthBody`	Length of the text of the body (in characters); no value if claim is single-line format and body does not exist (or not converted)	integer
`isMeansPreamble`	Does preamble use a means-plus-function phrase?	0 = no; 1 = yes
`isMeansBody`	Does body use a means-plus-function-phrase?	0 = no; 1 = yes
`isMeans`	Is claim a means-plus-function claim?	0 = no; 1 = yes
`processPreamble`	Does preamble use terms "method" or "claim" (simple classifier)?	0 = no; 1 = yes
`processBody`	Does body use terms "method" or "claim" (simple classifier)?	0 = no; 1 = yes
`processSimple`	Does claim use terms "method" or "claim" either in the preamble or the body (simple classifier)?	0 = no; 1 = yes
`claimType`	Invention type of the claim (Table 2)	0 = no category; 1 = process; 2 = product; 3 = product-by-process
`preambleType`	Preamble type (Table 2)	0 = empty; 1 = process; 2 = product; 3 = by-process
`bodyType`	Body type (Table 2)	0 = empty; 1 = process; 2 = product; 3 = mixed
`label`	Preamble-body type combination (Table 2)	string

Table 7: Patent-Level Information

Variable Name	Description	Values
patent_id	USPTO patent number	string
claims	Number of independent claims; the sum of processClaims, productClaims, prodByProcessClaims, and noCategory	integer
noCategory	Number of independent claims without a claim type (claimType= 0)	integer
processClaims	Number of process claims (claimType= 1)	integer
productClaims	Number of product claims (claimType= 2)	integer
prodByProcessClaims	Number of product-by-process claims (claimType= 3)	integer
firstClaim	claimType of the first independent claim of the patent	integer
simpleProcessClaims	Number of process claims by simple approach (processSimple = 1)	integer
simpleProcessPreamble	Number of process claims by simple approach, preamble only (processPreamble = 1)	integer
meansClaims	Number of means-plus-function claims	integer
meansFirst	Is first claim a means-plus-function claim?	0 = no; 1 = yes
JepsonClaims	Number of Jepson claims	integer
JepsonFirst	Is first claim a Jepson claim?	0 = no; 1 = yes
lengthPreambles	Average length of the text of the preambles of multi-line claims (in characters); singleReformat= 0, = 1, or = 2; no value if singleReformat= 3	numeric
lengthBodies	Average length of the text of the bodies of multi-line claims (in characters); singleReformat= 0, = 1, or = 2; no value if singleReformat= 3	numeric