Patent Claim Structure Recognition

René Hackl-Sommer and Michael Schwantner

Abstract In patents, the claims section is the most relevant part. It is written in a legal jargon, containing independent and dependent claims, forming a hierarchy. We present our work aimed at automatically identifying that hierarchy within complete patent claim texts. Beginning with a short introduction into patent claims and typical use cases for searching in claims, we proceed to show results from a preliminary context analysis with English claims from the European Patents Fulltext (EPFULL) database. We point out some possibilities with which claim dependency is indicated in the text and show a way of identifying them. Additionally, we describe several of the problems encountered, in particular problems resulting from noisy data. Finally, we show results from our internal evaluations, in which accuracies greater than 93% were measured. We also indicate areas of further research.

René Hackl-Sommer
FIZ Karlsruhe, Herrmann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen,
✉ rene.hackl-sommer@FIZ-Karlsruhe.de

Michael Schwantner
FIZ Karlsruhe, Herrmann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen,
✉ michael.schwantner@FIZ-Karlsruhe.de

ARCHIVES OF DATA SCIENCE, SERIES A
(ONLINE FIRST)
KIT SCIENTIFIC PUBLISHING
Vol. 1, No. 2, 2017

DOI 10.5445/KSP/1000058749/17
ISSN 2363-9881

1 Introduction

Patents have a double function. On the one hand, a granted patent gives the owner for a certain time, usually for 20 years, the right to exclude others from making or bringing to market the described invention. In return, the applicant has to describe his invention in such detail, that this enables '*any person skilled in the art or science to which the invention or discovery appertains, or with which it is most nearly connected, to make and use the same*'[1].

This double function is reflected in the two essential textual parts of the application: The *detailed description* describes the invention, and the *claims* '*define the matter for which protection is sought*'[2]. The description is the part of the patent which is written in a style similar to scientific papers. It may be longer than these and often contains examples of the invention which help a skilled person to understand the content. The claims constitute a legal text, describing the details of the invention. They have to be crafted with special diligence; a formulation too narrow could unnecessarily limit the protection. For this reason, they are usually authored by patent agents for other patent experts and thus written in a legal language.

Patents are a vital economic factor and constitute important assets of a company. In the industry, questions like '*Has the technical innovation X already been described in a patent?*' are often of fundamental importance and a considerable effort is expended to get a correct answer. Translated into the terminology of information retrieval, this means that for a high recall a lower precision is accepted. To answer the questions mentioned above, many companies employ specialised information professionals with patent retrieval being one of their main tasks. They use the document structure to restrict their queries to the claims section where appropriate. But the claims themselves build up a structure which could be used for an even more precise retrieval, if this structure were explicitly available.

To begin with, claims can be differentiated into *independent* and *dependent* claims. Describing the matter of invention, the claims start with the general and end with the specific. Dependent claims refer to independent claims and are narrower in scope. They further qualify what has already been claimed. A

[1] http://mpep.uspto.gov/RDMS/detail/manual/MPEP/current/d0e18.xml#/manual/MPEP/current/d0e42440.xml

[2] The European Patent Convention, Art. 84, http://www.epo.org/law-practice/legal-texts/html/epc/2013/e/ar84.html

claim may refer back to a single previous claim or to several previous claims. An example would be:

1. A composition comprising [...] alkenyl succinic anhydride substituted starch, [...]
3. The composition of claim 1 wherein the alkenyl succinic anhydride substituted starch is n-octenyl succinic anhydride (OSA) substituted starch.
4. The composition of claim 3 wherein the alkenyl succinic anhydride substituted starch is n-octenyl succinic anhydride (OSA) substituted waxy maize starch.

As can be seen, the claim dependencies form a hierarchy; thus the whole set of claims can be considered as a directed graph.

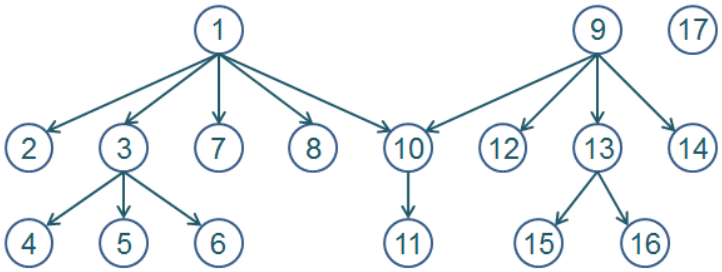

Fig. 1 Dependencies in a claim set.

Figure 1 shows an example of a directed graph for a claim set of 17 claims with three independent claims (nos. 1, 9, 17) and 14 dependent claims. Claim no. 10 has a multiple dependency (from claims no. 1 and 9). Another relevant concept in this context is the claim group. It consists of exactly one independent claim and all claims which are directly or indirectly dependent from this claim. The set in Fig. 1 has three claim groups:

- claim nos. 1-8 and nos. 10-11;
- claim nos. 9-16;
- claim no. 17.

A claim may belong to more than one group, and a group may consist of only one claim. To make these structures available during retrieval has several benefits for the user. If the user can limit his search to the independent claims, he will get mostly hits where the concepts covered by his search terms are of fundamental importance. Limiting the search to a claim group and using two or more search phrases will find mainly patents where these phrases are used

in a close semantic relation. Thus these two features would allow for a more precise search. Additionally, visualising the hierarchies with a diagram as in Fig. 1 allows for a faster understanding of the scope and coverage of the claims.

In this paper, we describe our approach to automatically identify the claims' structure. After introducing related work, we outline the methods used to collect an as exhaustive as possible set of phrases which are useful for identifying claim dependency. We then detail the workflow used for, first, separating the individual claims from each other, and then resolving the full claim hierarchy. We conclude the paper with the evaluation results for our method and some remarks on issues for further development.

2 Related Work

Several institutions or companies have incorporated at least a partial recognition of the claim structure in their products or online services. Espacenet[3], the European Patent Office's public search engine, offers a tree like graphic representation of a patent's claim structure in their viewer. The french software company Intellixir[4] offers the detection of independent claims as part of their analysis portfolio. The LexisNexis product TotalPatent[5] also incorporates a visualisation of claim hierarchies. However, to our knowledge none of these bodies have yet published any papers on their work.

Apart from these, there has been research in recent years into patent information retrieval in general and also more specifically into patent claims. Beginning in 2007, the now-extinct Information Retrieval Facility[6] staged a series of conferences, which focused on patents. Also, the EU-financed projects PATEXPERT[7], TOPAS[8], and iPatDoc[9] united a host of partners to further our understanding of patent searching and analysis as well as to help develop tools for searchers.

[3] http://worldwide.espacenet.com/

[4] http://intellixir.com/

[5] http://www.lexisnexis.com/en-us/products/total-patent.page

[6] http://www.ir-facility.org/

[7] project website expired, cf. Wanner et al (2008)

[8] http://topasproject.eu/, cf. Brügmann et al (2015)

[9] http://www.ipatdoc.eu/

A linguistic perspective is dominant in the work of Sheremetyeva (2003), who looks at structure in patent claims from a sentence and sub-sentence level. She attempts to break up complex nominal phrases into smaller units, which would be easier readable and understandable. A similar approach is described in Shinmori et al (2003b), who are investigating the main claims of Japanese patents to generate meaningful fragments, with the added complexity of the Japanese language, in which the combination of kanji characters in patent claims often is unique to patents.

The topic of improving claim readability and understandability is continued in Shinmori et al (2003a), Shinmori and Okumura (2004), and Parapatics and Dittenbach (2011); structural parsing related efforts are reported in Verberne et al (2010), D'hondt et al (2011), and Yang and Soo (2012). In the latter approaches, dependency parsers are used. These are, however, parsers aimed at finding grammatical relations in claims. They are not parsers aimed at identifying which claims depend on which other claims.

Claim dependency does play a central part in Lopatecki (2008). The author analyses German and US American patents and investigates two hypotheses. The first hypothesis:

> 'The occurrence of references in patent claims is a direct indicator to identify and separate independent from dependent claims.' (Lopatecki, 2008, p. 55)

To exemplify, if a phrase like '*according to claim 1*' is present in a claim, it is a dependent claim. If such phrases are absent, it is safe to conclude that the given claim is independent.

The second hypothesis:

> 'The well-structured language of patent claims allows for structure-based parsing of the claims to identify references.' (Lopatecki, 2008, p. 56)

While Lopatecki confirms the viability of both hypotheses, we would – in an attempt to better differentiate from linguistic approaches like the ones referenced above – prefer to paraphrase the second one as:

> The formulaic language of patent claims allows for pattern-based analysis of the claims to identify references.

We believe that both hypotheses are good starting points. However, the amount of heterogeneity that Lopatecki reports is very low and does not reflect our experiences with the full spectrum of patent texts. In our work, we identify more heterogeneity, like semantic variations of the term '*claim*', and take common sources of noise, like missing or misplaced blanks, into account.

We do also show a way of analysing text to identify relevant terms to improve the coverage in the following section.

3 Context Analysis

An impression of the formulaic use of language in patent claims can be gained by reading a small number of claims. It will quickly become evident that some phrases – with slight variations – occur often, e.g. '*according to claim 1*', '*of claim 2*', or '*of any preceding claim*'. This can be partly explained by jurisdiction-dependent regulations governing claim dependency construction. Yet, we cannot rely on the specific limits that the regulations are imposing as we aim to cover all English claims regardless of their origin: What is allowed in one jurisdiction might be forbidden in another one.

Thus, to establish a pattern-based approach, the different phrases and variations need to be known as comprehensively as possible. The patent experts at FIZ Karlsruhe provided us with many of the most frequent terms and phrases used for claim dependency construction. These phrases constituted our first set of claim spelling variants (CSV). Intermittent evaluation of our first prototype then revealed that there appeared to be a significant number of derivations and alternative formulations which were yet unconsidered.

In order to close this gap in a structured and thorough manner, we performed a context analysis with a large amount of patent claim data. More precisely, we extracted all English language patent claim sections from the EPFULL (European Patents Full-Text) database (2.4 million claim sections). We split these claim sections up into individual claims, as per their XML markup. The individual claims amounted to 24.3 million, and about 25.000 patents had more than 50 claims. We indexed these with Elasticsearch (`https://www.elastic.co/`) and also performed most of the subsequent analysis with the associated API. Part of the analysis was performed with Lucene's (`https://lucene.apache.org/`) own HighFreqTerms library.

Then we conducted a keyword-in-context analysis (KWIC) on the basis of the CSV terms. We evaluated the hit locations for the query terms in all of the documents and counted the term occurrences. The intellectual analysis of the results revealed many variations stemming from missing blanks, as in '*toclaim*', '*inclaim*', or '*claim2*'. We added these terms either to our CSV set, or, in the cases of numbers, we adapted our patterns used for matching to cover

such instances. With this new set of CSV terms we did a KWIC analysis which showed that the three most frequent tokens following a CSV term were numbers, commas, and the term '*wherein*', which is unsurprising because formulations like '*according to claim 1*', '*any of the previous claims, wherein*' abound in patent claims. Less frequent tokens which were useful included '(NUMBER)' for any number (see Sect. 4.3), 'NUMBERor', and 'l' and 'I'. The latter two characters are occasionally found instead of the numeral '1'. This would be more expected with data that originates from OCR scans, but nevertheless we encountered roughly 32.000 claims of this type.

In another step, we investigated the terms preceding a CSV term (we call them *predecessor terms*). This includes phrases like '*one of the* [claims 3 to 5]' or '*any one of the previous* [claims]'; the analysis of predecessor terms yielded mostly adverbs like '*aforementioned*' or the frequent spelling error '*preceeding*'.

Finally, we carried out one last KWIC analysis. This time, we evaluated the hit locations of all the previously identified predecessor terms and investigated the positions directly after these terms. Clearly, many of the successor terms found were already present in our CSV set. But there were also others like '*reference claim*' and especially frequent typos like '*calim*' or '*cliam*' which we added to the CSV and thus rounded off our coverage.

In summary, the context analysis shows plenty of evidence for the formulaic nature of patent claim composition. This formulaic nature helps reduce the complexity of the language in patent claims as far as claim dependency is concerned. Furthermore, our findings support the assumption that a pattern-based approach towards extracting dependency information is not only feasible, but that such an approach can also be expected to perform well.

4 Claim Structure Recognition

4.1 Workflow

The workflow for the identification of the complete claim hierarchy is straightforward. The following four main aspects are important.

1. Claim segmentation: A method needs to be devised that detects the beginning of new claims, thus segmenting the claim text.

2. Claim number recognition: Claims are numbered sequentially, starting with claim 1. For each claim, its number needs to be identified.
3. Claim categorisation: Each claim has to be categorised as either dependent or independent.
4. Claim dependency: In the case of dependent claims, the parent claims need to be fully extracted.

The following sections address these items as follows. We first examine the segmentation of a full text of claims into individual claims. Items 1 and 2 go hand in hand, as the claim numbering plays an important part in the segmentation of claims. While the task appears simple at first glance, there are some pitfalls involved which we report on. Next, we look at ways in which dependency can be expressed and use these to provide methods regarding items 3 and 4. In the last section, we shed some light on the inner workings of the implementation.

4.2 Claim Segmentation

To identify individual claims, a reliable method to find claim beginnings is required. A claim can then be said to cover all text from one beginning to the next, including the first and excluding the latter. And in many cases, the beginnings of new claims are simple to identify:

- They begin with a new line.
- They contain an ordinal, which may be preceded by the term 'Claim' as in 'Claim 1'.
- The ordinal is followed by a number of characters: a blank, dot, hyphen, or closing parentheses; e.g. '1 ', '1.', '1-', or '1)'.
- Finally, there is a capital letter indicating the beginning of a new sentence.

The examples up to this point in the text followed these patterns and our first pass algorithm makes use of it as well. We have included further improvements in that algorithm to handle additional line-breaks and spacing variants, e.g. '*Claim 1 3. A method...*'.

In less frequent cases, the beginnings of new claims are not separated by line-breaks; sometimes, all claim text is even contained in a single claim *en bloc*. Here, the algorithm depicted above falls short and we need to look further into the text. This involves removing the requirement for line-breaks while adding

awareness for the last found claim number. In that manner the algorithm knows which claim number is expected next and can attempt to find it.

In some rare instances, claims contain passages which obey said rules, but which do not represent a sequence of claims, e.g. recipe-like instructions in the chemical domain:

2. Procedure to obtain [...] **including the following steps**:
    1. Provision of the powder mixture in suitable proportions [...]
    2. Mixture with tap water and stirred until dissolved.
    3. Addition of the additive [...]
3. Procedure to obtain [...]

Spaces or tabs are not guaranteed to be present in such cases. Here, the algorithm will look for claim 3 and the section beginning with '*3. Addition*' is a strong candidate, however, it is part of a sequence of steps. Subsequently, the real claim 3 appears. The algorithm then has to make a decision which one to keep. We decided to consider the first occurrence as the real claim, and write further occurrences into a logfile for later analysis.

4.3 Claim Dependency Analysis

Once the individual claims are identified, we determine whether a given claim is dependent or independent. The location of the dependency information within the claim text is irrelevant for identification purposes. Often, the dependency information is closer to the beginning of the claim. However, in some claims the dependency information is given at the very end of the claim. Usually it is provided in one sequence of varying length. It is possible for dependencies to occur in multiple places in a claim, though.

Technically speaking, we are building on the following annotations for our analysis:

- `PREDECESSOR_TERM`: e.g. 'any one of the', 'one or several of'. To facilitate pattern processing as described below, `PREDECESSOR_TERM`s are further subdivided into different types. E.g. 'the' forms the valid sequence 'the first claim', but 'any of the first claims' is not a valid sequence. Because of space limits, we are not discussing the typing in detail.
- `CLAIM_STRING`: as shown above, the annotation marks occurrences of 'claim' and its variants, including irregular ones like 'claim1' where a blank is missing (flagged).

- NUMBER: numbers, either in digits or spelled-out, as well as cardinals or ordinals, e.g. 'according to claim 1', 'according to claim one', 'according to the first claim'. Ordinal numbers are identified up to 10. The hard stop at 10 has been put in place, because the frequencies for ordinal numbers decrease rapidly and it does not make sense to put resources into attempting to find very rare occurrences (see Table 1, where we report frequencies of phrases for three different patent databases).

Query	EPFULL	PCTFULL	GBFULL
first claim or 1st claim	596	7995	3473
second claim or 2nd claim	76	997	903
third claim or 3rd claim	48	159	362
fourth claim or 4th claim	31	24	180
tenth claim or 10th claim	8	5	4

Table 1 Frequencies of ordinal numbers in claim references.

- INTERVAL: intervals are expressed with a limited number of phrases. Common examples are 'between x and y', 'x to y', and 'x-y'.
- FILLER_ADVERB: these adverbs, like 'preceding', 'previous', or 'before', may occur before or after a CLAIM_STRING. A flag is set to indicate valid positions. Sometimes, their presence is required to form an acceptable dependency reference, e.g. 'according to any claim' would probably not pass examination, however 'according to any claim before' would.
- ENUMERATION: e.g. 'according to claim 1, 2, or 3' or 'according to claims 1, 3 to 5 and 10-20'

With these annotations, we can construct sequence patterns and implement case-by-case handling for them, for instance:

- PREDECESSOR_TERM CLAIM_STRING INTERVAL:
  e.g. 'one or more of claims 1 to 5'
- PREDECESSOR_TERM FILLER_ADVERB CLAIM_STRING:
  e.g. 'the previous claim'
- PREDECESSOR_TERM FILLER_ADVERB CLAIM_STRING NUMBER:
  e.g. 'the previous claim 6'

The use of flags is an important tool to mitigate noise. In Sect. 3 we had shown how other characters can be found instead of the numeral '1'. We also know that blanks are frequently missing. To provide best results, we need to

take this into account for all patterns. The last pattern above also matches 'the previous claiml 6', a flag indicates the irregularity of the `CLAIM_STRING` and we correctly process the information as if the string had read 'the previous claim 16'.

4.4 Implementation

We are accessing the raw data in XML, and while the XML is well-formed, there is still considerable divergence in the availability of the claim texts. As we want to devise a method that is capable of analysing it all, our general approach decouples the mere text extraction from the raw data from the analysis steps. For the extraction we have devised a software module in Java. It processes the XML data removing all mark-up and unifying the divergent content to deliver a consistent view of the data. While most of this procedure is only a technicality, the one critical aspect is the presence or absence of line-breaks. In the raw data, claim text can be present with line-breaks separating individual claims:

"1. A method for the classification of textual data, ... ¶
2. The method of claim 1, ..."

Unfortunately, the claims do also occur as *en bloc claims*:

"1. A method for the classification of textual data, ... 2. The method of claim 1, ..."

Lastly, mixes of the above, where some claims in a patent are separated individually, while others are present *en bloc*, are possible, too. It is for this reason that the sole analysis of the XML structure would be insufficient to provide good coverage for the claim structure. For best results, we have to look into the full text.

For analysis, we are resorting to the Apache UIMA framework (UIMA = Unstructured Information Management Architecture, `http://uima.apache.org/`). This is a versatile software which, as a key feature, offers a pipeline. Text is inserted at the beginning of the pipeline, then analysis steps are carried out with so-called *analysis engines*, which in turn are producing *annotations*. These annotations are available to further analysis engines. For example, given the text

```
6. The method according to claims 1 to 5, ...,
```

one analysis engine identifies the numbers 6, 1, and 5 and consequently assigns NUMBER annotations. Also identified are CLAIM_STRING annotations. In UIMA, the annotations are represented in a superordinate layer:

```
NUMBER                      CLAIM_STRING    NUMBER     NUMBER
6       . The method [...] claims          1       to 5
```

Here, a second analysis engine identifies the pattern 'NUMBER to NUMBER', simultaneously taking into account annotation data and text data, and assigns an INTERVAL annotation:

```
                                             I N T E R V A L
NUMBER                      CLAIM_STRING    NUMBER     NUMBER
6       . The method [...] claims          1       to 5
```

In that manner, annotations can be 'stacked' on top of each other.

5 Evaluation

As part of our implementation we ran evaluations over all builds of our prototype. Beyond that, we performed two rounds of intellectual evaluations. These are time-consuming, because the evaluators need to read and understand the full claims text to be able to compare them to the output. To provide support in this area, we generated a text-based output in a collapsible tree structure and also a graph-based output. While the graph was visually appealing, it stopped being helpful for claims with many multiple dependent claims.

For evaluation purposes, we extracted random documents from our repository, performed the analysis, and delivered the results to in-house experts from the sales department. After each round of evaluation, we analysed the feedback gathered and further improved the prototype system. Because we had experienced more difficulty with patents containing many claims, we chose to over-represent these.

Dataset 1

- 100 random documents with at least one claim
- Additional 30 random documents with at least 50 claims
- Accuracy of claim segmentation: ~99% (129 of 130)

- Accuracy of claim dependency recognition: ~93% (122 of 130)

The accuracy is the percentage of correctly processed documents. A document is considered correctly processed, if *all* its claims and hierarchies are interpreted correctly. For a human reader the claim dependencies and hierarchies are unambiguous so problems of inter-annotator agreement do not arise here. Initially, we considered the possibility to generate more sophisticated measures that would allow us to gain insights into errors by category. However, when the results from the first evaluation round came in, this idea was dismissed. The results in the high 90ies already exceeded our minimum target requirement of 85%. Also, such a low number of erroneous documents can easily be analysed manually.

Dataset 2

- 100 random documents with at least one claim
- Additional 30 random documents with at least 50 claims
- No overlap with Dataset 1
- Accuracy of claim segmentation: ~99% (129 of 130)
- Accuracy of claim dependency recognition: ~96% (125 of 130)

In our case, one major cause of recognition failures were mixes of enumerations and intervals. They do occur in many forms, and all need to be represented in the algorithm for successful identification. Same examples:

- any one of claim 1 to 5, 7, 8 or 10, wherein
- any one of claims 25 to 34 or 40 to 44 or 52 to 55
- one or more of claims 1 and 5 to 7,
- any of cliam1- 3 and 5

The last example illustrates the need to be able to handle noisy data, especially of the missing or extra blank type. Our system can successfully gather the dependency information from such data.

Self-referencing claims are a persisting problem area. These are claims which refer to their own claim numbers in a dependency formulation, e.g. '*12. The process of claim 12, wherein...*'. Usually, the number at the beginning of the line is correct, but the dependency information should refer to an earlier claim. It appears, that the real dependency can to date only be reliably established by intellectual analysis.

In total, we have evaluated a sample size of 260 documents out of 2.4 million patents. The accuracy of the claim dependency recognition in the sample is

95% (247 of 260). Under the conservative assumption that errors occur in 5% of documents, the results are statistically meaningful and reach above the 95% confidence level.

6 Conclusion

The recognition of the claim structure in patents is a task that is well suited for regular expression based text mining, confirming Lopatecki's second hypothesis (see Sect. 2). We have indicated many real-world areas in which variability may be found and have shown a way to identify such variability. As is usually the case, the correct recognition of the last few percentages of problematic cases is getting more difficult and more resources have to be expended for less gain. Considering our initial minimum quality requirement of more than 85% though, we have comfortably surpassed this threshold. Most of the remaining mistakes occur within the subtask of recognising the dependency information in a claim accurately and completely. We can perform the subtask of identifying independent claims with close to certainty. Thus, we have also confirmed Lopatecki's first hypothesis and finding, that claim independence can be constituted by absence of claim dependence information.

We note that the expansion of the shown methods to other languages is an area of further research. The identification of the object of a claim, e.g. a composition, a device, or an apparatus, would also be interesting. It could then be determined if the object changes within a claim chain. For instance, a claim that is dependent on a compound claim could introduce a composition as the new object. Such information would be relevant to users. Also, the integration of the full claim dependency hierarchy, the claim chains, for use in online searches in combination with regular full-text searches is challenging. However, being able to limit searches to independent claims and claim groups, instead of needing to search the full claims text, is already a noticeable improvement and will help boost high precision searches.

References

Brügmann S, Bouayad-Agha N, Burga A, et al (2015) Towards content-oriented patent document processing: Intelligent patent analysis and summarization.

World Patent Information 40:30–42

D'hondt E, Verberne S, Alink W, Cornacchia R (2011) Combining document representations for prior-art retrieval. In: CLEF (Notebook Papers/Labs/Workshop)

Lopatecki L (2008) Comparative analysis of German and American patent claim text structures as a basis for the evaluation of patent scope. Master's thesis, University of Hildesheim, Germany

Parapatics P, Dittenbach M (2011) Patent claim decomposition for improved information extraction. In: Lupu M, Mayer K, Tait J, Trippe AJ (eds) Current Challenges in Patent Information Retrieval, The Information Retrieval Series, vol 29, Springer Berlin Heidelberg, pp 197–216

Sheremetyeva S (2003) Natural language analysis of patent claims. In: Proceedings of the ACL-2003 workshop on patent corpus processing, Association for Computational Linguistics, Morristown, NJ, USA, pp 66–73

Shinmori A, Okumura M (2004) Aligning patent claims with detailed descriptions for readability. In: Proceedings of the fourth NTCIR workshop on research in information retrieval, automatic text summarization and question answering, National Institute of Informatics, Japan

Shinmori A, Okumura M, Marukawa Y, Iwayama M (2003a) Patent claim processing for readability: Structure analysis and term explanation. In: Proceedings of the ACL-2003 workshop on patent corpus processing, Association for Computational Linguistics, Morristown, NJ, USA, pp 56–65

Shinmori A, Okumura M, Marukawa Y, Iwayama M (2003b) Rhetorical structure analysis of japanese patent claims using cue phrases. In: Proceedings of the third NTCIR workshop on research in information retrieval, automatic text summarization and question answering, National Institute of Informatics, Japan

Verberne S, D'hondt E, Oostdijk N, Koster C (2010) Quantifying the challenges in parsing patent claims. In: Proceedings of the 1st international workshop on advances in patent information retrieval (AsPIRe 2010), pp 14–21

Wanner L, Baeza-Yates R, Brügmann S, Codina J, Diallo B, Escorsa E, Giereth M, Kompatsiaris Y, Papadopoulos S, Pianta E, et al (2008) Towards content-oriented patent document processing. World Patent Information 30(1):21–33

Yang SY, Soo VW (2012) Extract conceptual graphs from plain texts in patent claims. Engineering Applications of Artificial Intelligence 25(4):874–887