

特許権侵害の判定支援システムの実験的試作に関する一考察

安達ほのか^{†1} 平塚三好^{†2} 谷川英和^{†3}

特許権の侵害・非侵害の判断において重要なのは特許請求の範囲の解釈にある。しかしこの特許請求の範囲の解釈は難しく、正確な判断は専門家に任せる必要があるのが現状である。しかし特許権侵害を未然に防ぎ、正常な業務を行うためには企業側である程度判断をする必要がある。そこで本研究では、侵害・非侵害の判断手法をある程度公式化し、それをシステムとして落とし込むことによって、専門家でない人でも侵害の目安をつける手助けを行えるようにすることを目的としている。今回は試験的にシンプルな仮想事例を想定し、テキストマイニングなど自然言語処理の手法を活用し、特許の侵害・非侵害の判定支援の可能性を検証する。

A study on experimental prototype of the decision support system of patent infringement

HONOKA ADACHI^{†1} MITSUYOSHI HIRATSUKA^{†2}
HIDEKAZU TANIGAWA^{†3}

Claim Interpretation is important in the determination of infringement and non-infringement of patent rights. Though, this claim construction is difficult, it is necessary to leave to the experts accurate judgment. However, it is need to make a judgment to some extent by the companies in order to avoid the patent infringement, to perform smooth business. In this study, by the system of the judgment method of infringement and non-infringement, and to help to give an indication of infringement in layman. In this study, we assume a virtual simple case on a trial basis and validate the judgment of infringement and non-infringement of the patent or can not from the perspective of natural language processing such as test mining.

1. はじめに

事業を行っている者にとって、自社の製品が他の特許の侵害をしているかどうかは、重要な問題である。しかし、特許の専門家でない者にとって、公開されている特許公報から自社の製品に近いものを探し出し、その侵害の判定を行うことは難しいことである。そこで、現在は特許侵害の判定は特許の専門家に相談することが一般的となっている。しかし、すべての製品について事前に特許侵害をしていないか調査を行うことは、業務上大変な労力を強いられる。そこで、本研究では普段人間の手によって行われている侵害判定の手段を明確化し、それをシステムとして落とし込むことによって、専門家でない人でも侵害の目安をつける手助けを行えるようにすることにより、事業を円滑に進めるための支援を行うシステムの開発を最終的な目標としている。現在の人間の手による侵害の判断手法と、その中で本研究が目的とするシステム化の範囲を図1に示す。

そのシステム構築の前段階として、本研究では主に特許の明細書に記載されている「特許請求の範囲」に書かれている文言に着目し、侵害の判定を行うことができないか検

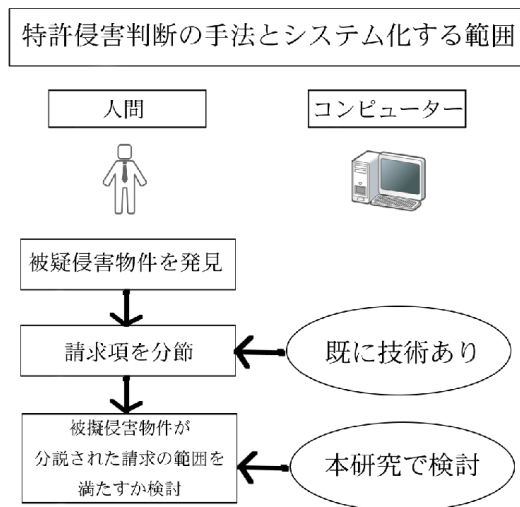


図1 システム化の範囲

討する。また、この特許請求の範囲に書かれている文言に着目する手法の前提条件として、侵害しているか確かめたい物件（以下、被疑侵害物件と言う）も、その技術的範囲を文言として書かれていることを前提とする。実務的には被疑侵害物件の技術的範囲が文言として記載されていることは滅多にないケースであるが、あくまでもシステム構築のための実験的試作なので、ここでは便宜上文言が存在するとする。

なお、特許の侵害には権利者以外の者が正当な理由や権

†1 東京理科大学
Tokyo University of Science

†2 東京理科大学
Tokyo University of Science

†3 有限会社アイ・アール・ディー
IRD Corp.

利がない状態で特許発明を業として実施する「直接侵害」と、本来的には特許権侵害ではないが、法上侵害とみなされる「間接侵害」が存在するが、本研究では直接侵害を想定している。また、均等論については考慮しない。

2. 特許権侵害の判定手法

特許権の侵害判定では、まず特許請求の範囲に書かれている文章を分節し、上から順にA, B, C …とつけていく。例えば、ある鉛筆があって、その特許請求の範囲を、

- A. 黒鉛と粘土からなる芯を
- B. 断面六角形状に形成された軸の長手方向に
- C. 密着させて被覆したことを特徴とする鉛筆

と分節できたとする。この分節された文章（以下、構成要素と呼ぶ）の満ち欠けによって、侵害は判定される。以下に例をいくつか示す。

2.1 侵害の例

(1) 外的付加

構成要素A, B, Cすべてを充足し、更に新たな要素Dを加えた物。

例：鉛筆に消しゴムを追加した鉛筆

(2) 内的付加

新たな要素を加えたという点では外的付加と一緒に、構成要素の一部を限定する形となっている物。

例：軸の素材が紙であったような場合

(構成要素Bを限定している)

2.2 非侵害の例

(1) 欠缺

構成要素A, B, Cのうちいくつかは揃っていない物。

例：芯が黒鉛のみからなる場合

(構成要素Aが欠缺している)

3. 仮想事例の作成

システムを作成する上で、システムが出力した結果が実際の結果とどう対応しているか検証するために、上に示した侵害のパターンを基にサンプル事例を用意した。今回の事例はできるだけシンプルなもの、侵害の判定が一目で分かりやすい本棚を対象として作成した。図2に示すもの

が今回の対象物件とする本棚である。この本棚は、側面に複数の穴が開いてあり、好きな高さの穴にネジなどの固定具を差し込むことによって、好きな高さに仕切り板を調節できるようにしたものである。

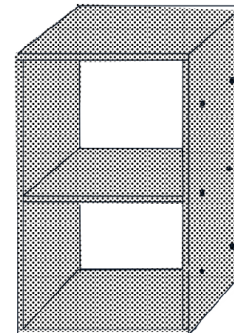


図2 対象物件の本棚

この本棚の特許請求の範囲は、以下のように記述した。

- A: 複数の開口部が設けられた第一の縦板と第二の縦板と、
- B: 縦板の最上部及び最下部に接続された横板と、
- C: 書籍を支持するための一枚以上の支持板と
- D: 前記開口部に挿入される挿入部を有する留め具
- E: とからなる本棚。

これに対し、この対象物件を侵害している事例を二つ（外的付加と内的付加の例）、侵害していない事例を一つ（欠缺の例）を用意した。それらを図3に示す。

まず、侵害である外的付加の例（図3の①）では、背面に新しい要素である裏板を付け加えることによって要素を付加した。特許請求の範囲の文言としては、構成要素BとCの間に新たな構成要素F「F：背面に前期縦板と前記横板とに垂直に接続された裏板と、」という要素を付け加えた。

次に、侵害の内的付加の例（図3の②）では、本棚の素材を「ベニヤ板」と限定することにより、「ベニヤ板」が「板」の下位概念に含まれ、要素を内的に付加した。文言では、構成要素A, B, Cに含まれている「板」という言葉をすべて「ベニヤ板」に置換し、それぞれA1, B1, C1と置き換えた。

更に、非侵害の欠缺の例（図3の③）は、縦板にあるネジなどの固定具の挿入部を削除し、仕切り板が完全に固定されるような本棚に限定した。挿入部と留め具を削除することで要素を減らし、侵害を防いだ。文言としては、構成要素Aの「複数の開口部が設けられた」という記述を削除しA1と置き換え、構成要素D「前記開口部に挿入される挿入部を有する留め具」の記述を丸ごと削除した。

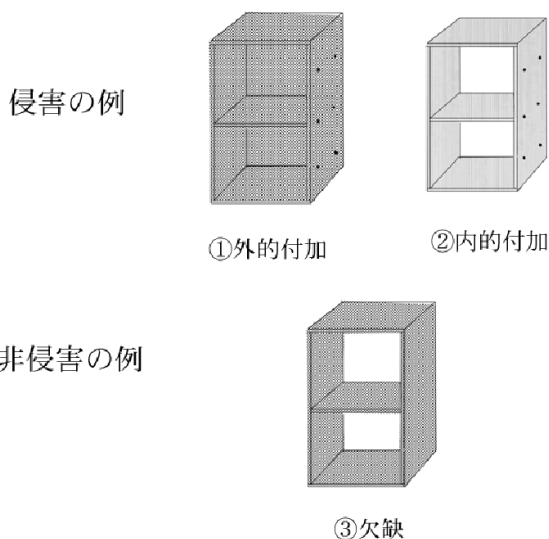


図3 サンプル事例

- A: 複数の開口部が設けられた第一の縦板と第二の縦板と、
- B: 縦板の最上部及び最下部に接続された横板と、
- C: 書籍を支持するための一枚以上の支持板と
- D: 前記開口部に挿入される挿入部を有する留め具
- E: とからなる本棚



- A [縦板] [開口部] [複数]
- B 「上部」「下部」「横板」
- C [支持板] [書籍]
- D [前記開口部] [挿入部] [留め具]
- E [本棚]

図4 単語の代入の方法

4. アルゴリズムの提唱

実際にシステムを作成するにあたって、システムのフローチャートを考察するとともに、それを実行するマシンの環境を述べてゆく。

4.1 プログラムの流れ図

全体の大まかな流れ図を図7のフローチャートに示す。フローチャートに沿って上から順番に説明してゆく。まず最初に、対象物件と被疑対象物件の特許請求の範囲の文章から重要単語を抜き出す。この作業について使用するソフトウェアについては後述する。そして対象物件の文章から抜き出した単語を、構成要素ごとに用意しておいた配列に保存する。これを図4に示す。(本棚の例でいうと、構成要素はA～Eまでの5つあるので、配列を5つ用意し、構成要素Aの文章から抜き出した単語は配列Aへ、構成要素Eから抜き出した単語は配列Eへそれぞれ格納してゆく。)次に、被疑侵害物件の文章から抜き出した単語について、上記配列に格納された単語と同じものがあるかどうかチェックしてゆく。この時、対象物件と同じように配列を用意しておき、同じ単語が現れた時は該当する構成要素の配列にその文字を代入し、もし違うものが現れた時は、その単語を別の場所に保存し一時的に覚えておく。配列を区別するため、最初に作成した配列を対象物件配列、後者を被疑対象物件配列と呼ぶ。外的付加の例で実際に試してみた例を図5に示す。

その作業を終えると、被疑対象物件の単語の中で、対象物件と使用されているものと同じ単語と違う単語か判別できる。図5の場合では、A'～E'の配列の中に入っている単語が対象物件と使用されているものと同じ単語であり、あぶれという配列に入っている単語が違う単語であることが分かる。この違う単語を、AからEのどの配列にも属さないことからあぶれ単語と呼ぶことにする。

次に、このあぶれ単語の中で、その単語の中に「前記」や「該」などといった言葉を含むものを除外する。これは、「前記」や「該」といった頭文字がついた単語は、それより以前に使われた単語を参照しているのであり、単語自体に新規性がなく、重要でないと判別したためである。実際、「前記」「該」といった文字を含む単語を除外せず試行した場合でも結果に差異は見られなかったので、単語の数を減らしシステムの負担を減らすことを優先した。

次に、あぶれ単語の中から、対象物件の単語に類似する関連語を探す。この手法で解析した例を図6に示す。

まず、あぶれ単語の関連語を、PSA (特許調査支援システム (PatentSearchAssistant)) [1] の関連語検索機能を用いて検索する。これは、単語の言い換えによる検索避けを避けるためである。

そしてあぶれ単語と得られた関連語を、形態素解析という文章を品詞に分解する技術を用いて分解する。この際に使用するソフトウェアについては次のセクションで後述する。

そしてその分解された品詞が、対象物件の単語を含んだ配列の中に含まれた単語がないか検索してゆく。図6にその一例を示す。外的付加の場合、あぶれ単語の中から「前

記「該」という言葉を除去すると、「裏板」「背面」の二つ

物件配列と、あぶれ単語を収納した配列と、下位概念の言

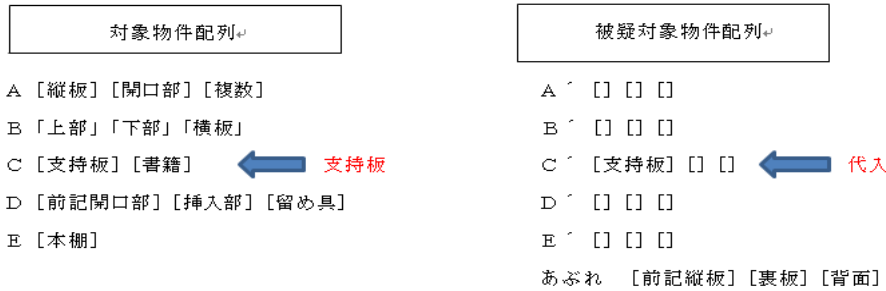


図5 被疑侵害物件配列への単語の代入の例

の単語が残る。これらを形態素解析にかけると、それぞれ「裏、板」「背面」のように分節される。次にこの品詞分解された単語それぞれで、これらの単語を含んだ単語が対象物件配列の中にないか探索する。例えば、「裏」という単語を含む単語は対象配列の中にはないが、「板」という単語を含む語は「縦板」「横板」「支持板」の3つが見つけられる。このようにしてすべてのあぶれ単語及びその関連語についてこの操作を行う。

そして、その操作において、分節された単語を含んだ単語が発見されたら、今度はその検索していた単語の分節する前の単語が、発見された単語の下位概念であるかどうかを判定する。

「板」であったら、分節される前の単語「裏板」が、それぞれ「縦板」「横板」「支持板」の下位概念であるかどうかを判定する。

しかし、この概念判定は現段階では機械が自動的に判別するには未だ難しい課題であり、現在そのシステム化のアルゴリズムは考案途中ではあるが、その精度に信憑性がなかったため今回の研究ではこの判定は人間の手によって行うこととした。その考案については今後の課題として考察で述べたい。

そこで具体的にどのように人間の手で概念判定を行っているかという、質問文を表示させ、それに「Y (yes)」か「N(no)」で返答を返すようにしている。具体的に「板」の例でゆくと、「板」を分節する前の単語は「裏板」であるから、この「裏板」がそれぞれ「縦板」「横板」「支持板」の下位概念であるか検証すればよい。

そして概念判定によってその検索していた単語が発見された単語の下位概念であると判定された場合には、別に用意していた下位概念単語配列に代入する。この配列に単語が代入されているか否かによって、後に下位概念の単語がなかったか確認する時の指標になる。逆に下位概念でないと判別された場合には特別な処理を行わず、次のあぶれ単語の分解された品詞で同様の操作を行う。これらの作業をすべてのあぶれ単語とその関連語について行う。

この過程をすべて終わると、対象物件配列と、被疑対象

被疑侵害物件：外的付加の場合の例

あぶれ単語 → 前記縦板, 前記横板, 前記開口部,
 裏板, 背面
 「前記」を削除後 → 裏板, 背面

形態素解析

裏板 → 裏 板
 背面 → 背 面

裏 (あるいはその関連語) を含む語 → なし
 板 (あるいはその関連語) を含む語 → 縦板 横板 支持板
 背面 (あるいはその関連語) を含む語 → なし

概念関係

「裏板」は「縦板」の下位概念ですか? → 「NO」
 「裏板」は「横板」の下位概念ですか? → 「NO」
 「裏板」は「支持板」の下位概念ですか? → 「NO」

図6 形態素解析・概念関係判定の例

葉を含んだ配列が存在することになる。そしてこれらの配列の中身の有無によって侵害判定を下すことになる。それにはいくつか条件の分岐がある。

その判定の条件は以下である。

条件① 被疑侵害物件配列に全く単語が代入されていない配列があるか否か

当てはまる場合は「欠缺」にあたり非侵害となる。そしてこの条件を満たしたものは第二の条件に移る。

条件② 下位概念である単語が一つでも存在するか否か

この条件に当てはまる場合は「内的付加」にあたるので侵害となり、当てはまらない場合は非侵害となる。

最後に全体を通してのフローチャートを示す。

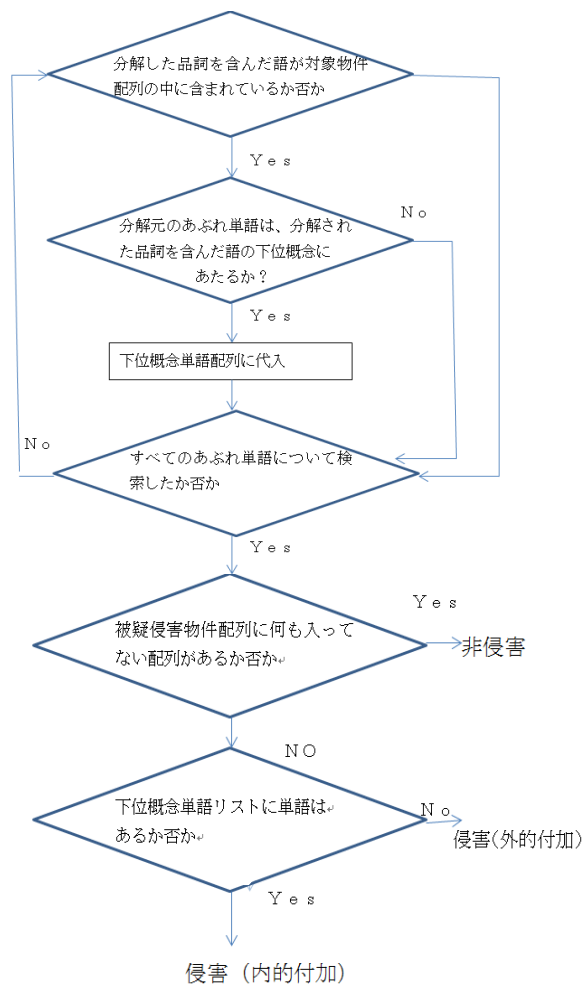
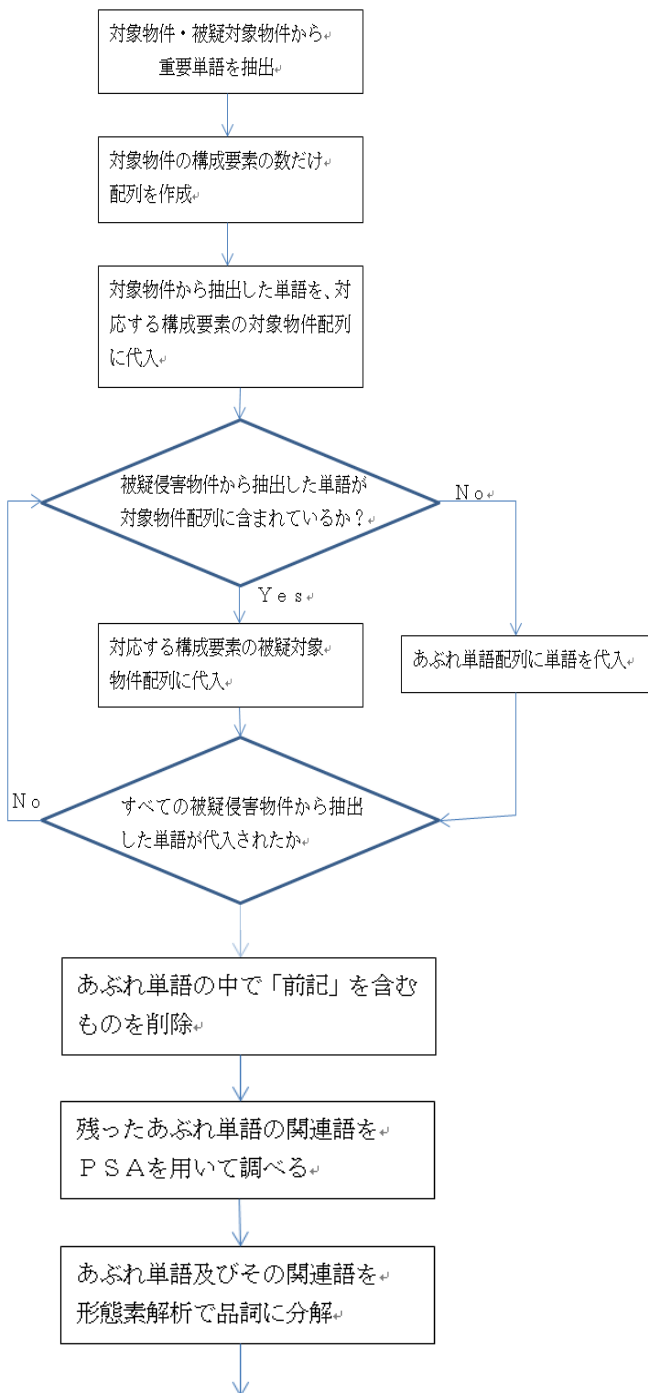


図7 フローチャート

4.2 実行環境

プログラミング言語は Perl 5 を用いて記述。使用 OS は Ubuntu 13.04 を Oracle VM VirtualBox [2] を用いた仮想マシン上で動かしている。ホストコンピュータは Window 7 を使用している。
 前述した特許請求の範囲の文章から重要単語を抜き出す作業は、ソフトウェア「TermExtract」[3] を使用した。同様に、形態素解析については、ソフトウェア「茶筌」[4] を使用した。

5. 検証結果

実行結果の一覧を表1に示す。

	正解	実行結果
外的付加	侵害	侵害
内的付加	侵害	非侵害
欠缺	非侵害	非侵害

表1 実行結果

6. 考察

予測と結果が相違した内的付加・変更について考察を行う。まず、内的付加について、途中経過を追跡してゆくと、まず専門用語を抽出し、あぶれ単語を抽出するまでは問題なく進んでいた。問題が生じたのは、形態素分析の過程においてである。

あぶれ単語には「縦ベニヤ板」「横ベニヤ板」「支持ベニヤ板」の三単語が抽出されていた。しかしこれらを形態素解析にかけると、「縦、ベニヤ板」「横、ベニヤ板」「支持、ベニヤ板」のように分節された。形態素解析の次の過程として、分節された単語を用いて更に適合する単語がないか検索する過程があるが、「ベニヤ板」では検索をかけても対象物件の単語を収納した配列に一致するものがなかった。

その結果、本来「板」という単語で検索して引っかかるはずの単語、関連語を検索することができなかった。そのため、概念関係の判定まで進む単語が存在しなかった。プログラムとしては、概念関係の判定まで進む単語がなかった時はその処理を飛ばし、その結果を「No」と返すように組まれていたので、結果として値が「No」と返ってしまった。本来は、「縦、ベニヤ、板」のように「ベニヤ」と「板」が分節されて抽出されていることを想定していた。そうすると、「板」というキーワードで検索することができ、それに基づき「ベニヤ板」というキーワードが抽出され、「ベニヤ板は板の下位概念に相当するか？」という概念関係の判定を行うことができた。そうすると、この文章を見た人間は「Yes」という値を返すことができ、正確な判定である「侵害」判定をすることができた。

これは、形態素解析を行うソフトウェア「茶筌」の処理の誤算が原因である。これを解決するためには、茶筌の挙動の再確認や、辞書ファイルの見直しが必要になってくるであろう。

7. 今後の課題

考察でも述べたように、まずは外的付加と変更の違いを判定する決定的事項を分析し、それをアルゴリズムに反映させる必要がある。そして次に茶筌の挙動をもう一度見直し、「ベニヤ板」を「ベニヤ、板」のように分節できるように検討する。もし不可能であれば、別の形態素解析のソフトウェアを試し、比較する。

そして、今回は実装を断念したが、概念関係の判定をできるだけシステム側で判定できるようなプログラムを考えていきたい。手法としては、上位下位概念関係を含んだデータベースを使用し、その中から近い意味を持つ単語を抽出し利用する方法を検討している。

また、難しいと推測されるという理由から、均等論の問題を今回は加味しなかった。最終的に侵害の判定の支援を行うのであれば、今後考えてゆく必要がある。

しかし、侵害・非侵害が曖昧なケースになればなるほど、最終的な判定などは人の手に頼る所が大きくなってくる。この研究の目的が特許権侵害の判定の支援を行うシステムの開発と銘打っているように、開発を進める上でどこまでがシステムの限界点であるのか、その線引きを見定めることも今後の課題としたい。

参考文献

- 1) 特許調査支援システム PatentSearchAssistant (オンライン), (<http://psa2.ip-engineering.jp/>)
- 2) Oracle VM VurtualBox (<https://www.virtualbox.org/>)
- 3) 専門用語 (キーワード) 自動抽出用 Perl モジュール "TermExtract" の解説 (<http://gensen.dl.itc.u-tokyo.ac.jp/termextract.html>)
- 4) ChaSen -- 形態素解析器 (<http://chasen-legacy.sourceforge.jp/>)
- 5) 浅野勝美 (2007) 特許権侵害の公式 / 商標権侵害の公式 産業科学システムズ
- 6) 牧野知彦 川田篤 高橋元弘 杉山一郎 堀電佳典 著, 東京弁護士会知的財産権法部 編 (2012) 特許・商標・不正競争関係訴訟の実務入門 商事法務