白梅学園大学・短期大学情報教育研究
2016, No. 19, 19－26.

剽窃検出ソフトWCopyfindの日本語文書への適用について

倉澤　寿之

インターネットの普及により、研究やレポート執筆において情報収集がしやすくなったが、このことは既存の情報を安易にcopy&pasteできるようになったということでもある。また、論文・レポート等をワードプロセッサで書くことが増え、レポート間のcopy&pasteも容易になってきているという現実がある。

他方、本学でもウェブベースの授業支援システム（dotCampus）が2014年度から導入され、学生のレポートを文書ファイルの形で集められるようになった。文書ファイルの形のレポートであれば、機械的に内容を検査することが可能になる。例えば、他学生のレポートをほとんど丸写ししたレポートを検出しようとすれば、印刷された紙で提出されたレポートの場合、記憶に頼る他にないが、文書ファイルであれば、記述内容を何らかの方法で評価することが可能になる。

こうした文書間の記述の重複を検出するソフトウェアとして、WCopyFind（Bloomfield, 2011）がある。英語など欧文をもとに開発されたものだが、多言語対応ということで、日本語文書も扱うことができる。ただ、実際に日本語文書を使った例は報告がないようである。本稿では、このWCopyFindの基本的な動作を踏まえた上で、日本語文書に適用する場合の留意点などをまとめておく。使用したWCopyFindのバージョンは、本稿執筆時点での最新版4.1.4である。

1．WCopyFindの基本動作（英文の場合）

WCopyFindの画面

図１がWCopyFindの画面である。以下、図１の画面の上から順に紹介しておく。

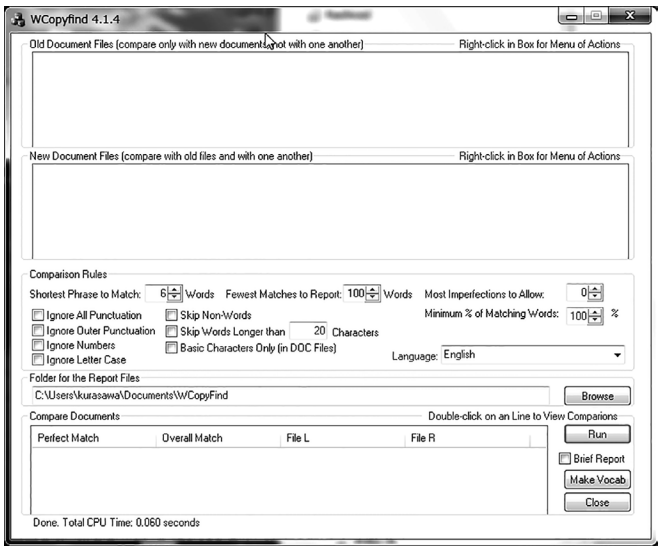


図１　WCopyFind画面

Old Document Files欄

文書ファイルを指定する欄の一つである。ここに挙げられた文書は、下のNew Document Files欄の文書との間で記述の重複がチェックされるが、ここに挙げられた文書相互間のチェックは行われない。引用文献や参考文献として挙げられる既存の論文等を指定することで、既存論文からの剽窃が行われていないかどうかを調べることになる。学生間でのレポート借用などをチェックするのが目的の場合、この欄は空欄で構わない。

New Document Files欄

文書ファイルを指定する欄であり、ここに挙げられた文書は、上のOld Document Filesとの間でのチェック、及びこの欄に挙げられた文書相互間でのチェックが行われる。チェック対象の文書を指定する欄である。

Comparison Rules

文書を比較する際の設定を行う。詳しくはこの後で紹介する。

Folder for the Report Files

比較結果を格納するフォルダを指定する。直接書き込んでもいいし、Browseボタンから選択することもできる。

Compare Documents

比較結果の履歴が表示される欄である。右のRunボタンで、文書の比較が実行される。

文書比較の設定項目

以下の設定項目の説明は、WCopyFindサイトの記述の翻訳に、筆者の検証結果を加味したものである。

Shortest Phrase to Match

一致したとみなす最小の単語数。例えばこれを６にセットすると、５単語以下のフレーズを一致とは見なさない。

推奨値は６。１から無限大まで設定可能。

Fewest Matches to Report

レポートする最小一致単語数。一致した単語数がこの数未満であれば、結果が出力されない。

推奨値なし。１から無限大まで設定可能。

Most Imperfections to Allow

許容される不一致単語の最大数。例えばこれを２とすると、WCopyFindは、途中に不一致単語が２つあっても、そのフレーズが一致したとみなす。この値を0にすると完全一致のみとなる。１～９とすれば曖昧さを残して一致を探すことができるが、この値を大きくすると実行が遅くなる。

推奨値は０から２。０から９まで設定可能。

Minimum % of Matching Words

フレーズが一致したとみなされるために必要な完全一致単語のパーセンテージ。100に設定すると、完全一致のみとなる。

推奨値は100（速さや完全一致を求める場合）から80（多少の曖昧さを残して一致を検出したい場合）。０から100まで設定可能。

Ignore All Punctuation

チェックすると、比較の際、すべての句読点を無視する。WCopyFindが生成する結果レポートには句読点が現れるが、一致判定には影響ない。句読点を無視すると一般に一致を検出しやすくなる。

本当にすべての句読点を無視したい場合以外は推奨されない。

Ignore Outer Punctuation

チェックすると、比較の際、単語の前後に現れる句読点を無視する。例えば「The box, which I found, is broken.」は「The box which I found is broken」と同じに扱われる。WCopyFindが生成する結果レポートには句読点が現れるが、一致判定には影響ない。句読点を無視すると一般に一致が起こりやすくなる。

完全な一致を望む場合にはチェックを推奨しないが、若干の変更がある場合にも一致を検出したければチェックを推奨。

※筆者の検証では、「Outer Punctuation」はカッコ、カンマ、ピリオドなどであり、「it's」のアポストロフィなど単語の間に入る記号は含まれない。

Ignore Numbers

チェックすると、数字を無視する。例えばこのパラメータをチェックすると、"8-fold"と"10-fold"は一致したとみなされる。WCopyFindが生成する結果レポートには数字が現れるが、一致判定には影響ない。数字を無視すると一般に一致が起こりやすくなる。

完全な一致を望む場合にはチェックを推奨しないが、多少の曖昧さを残して一致を検出したければチェックを推奨。

Ignore Letter Case

チェックすると、比較の際、大文字・小文字の区別を無視する。例えばこのパラメータをチェックすると、Wheneverとwheneverは一致したとみなされる。WCopyFindが生成する結果レポートには大文字・小文字の区別が現れるが、一致判定には影響ない。大文字・小文字の区別を無視すると一般に一致が起こりやすくなる。

完全な一致を望む場合にはチェックを推奨しないが、若干の変更がある場合にも一致を検出したければチェックを推奨。

Skip Non-Words

チェックすると、ハイフンとアポストロフィを除き、通常文字以外の文字を含む単語をスキップする。通常文字以外の文字を含む単語は一致判断に使われず、WCopyFindが生成する結果レポートにも現れない。このパラメータをチェックする場合は、（複数所有格など）句読点のついた単語がスキップされないように、"Ignore Outer Punctuation"にもチェックを入れるとよい。

完全な一致を望む場合にはチェックを推奨しないが、ファイル名、URL、ワープロ特有の記号など、文章以外の要素を多く含む文書の場合はチェックを推奨。

※筆者の検証では、この項目をチェックすると、記号が含まれる単語ばかりでなく、数字を含む単語、例えば「2nd」もスキップ（無視）されるようになる。

Skip Words Longer than [ ] Characters

チェックすると、指定した文字数より長い単語をスキップする。その単語は一致判断に使われず、WCopyFindの生成する結果レポートにも現れない。

実際に20文字以上の単語を含む文書の場合を除き、20程度の設定でチェックすることを推奨。チェックすることで、WCopyFindはファイル名、URL、ワープロ特有の記号など、文章以外の要素をスキップできる。

Basic Characters Only（in DOC Files）

Microsoft Wordの古い形式（.DOC）のファイルの場合にチェックすると、認識する文字セットを限定するようになり、WCopyFindは基本の文字セット以外を表に現れない文字とみなし一致判断に使わない。

非英文字があまり使われていない.DOCファイルを扱う場合にはチェックを推奨。

Language

文書に使用されている言語を指定する。適切に設定すると、WCopyFindが文字、句読点、表記法などを判断しやすくなる。

WCopyFindの動作

筆者の検証では、WCopyFindはスペースまたは改行記号で文を単語に分割し、連続して一致した単語が「Shortest Phrase to Match」の値以上になったとき、一致したフレーズを検出したことになり、その一致したフレーズに含まれる単語の総数が「Fewest Matches to Report」の値に達したとき、出力画面に現れる。50語ほどの簡単な英文を２つ用意して試した出力画面を図２に示す。

図２の「42」という数は、一致したと判断された箇所（フレーズ）に含まれる単語の数を表している。「82%」は単語として認識された51単語の中に占める一致箇所の単語数の割合である。この場合、２つの英文はほとんど同じで一部変えているにすぎないため、File LとFile Rの数字が同じになっているが、一般的にはそれぞれのファイルに含まれる単語数は異なるので、パーセンテージは変わってくる。「Perfect Match」と「Overall Match」も数字が同じであるが、これも「Most

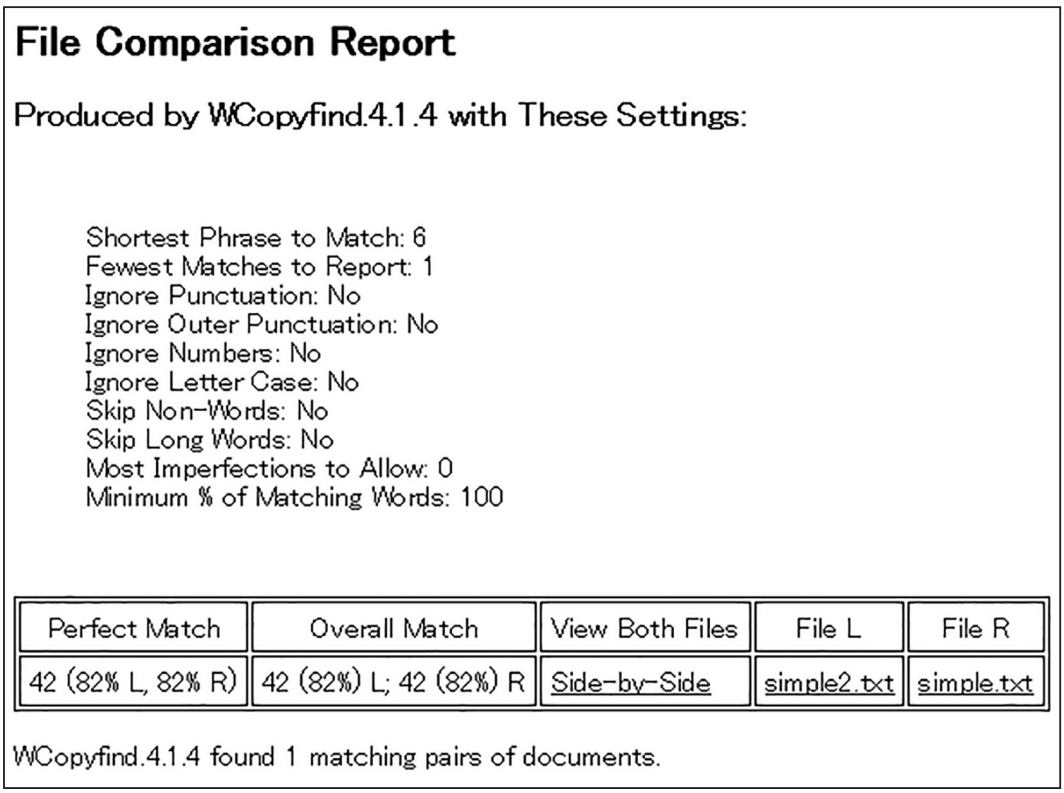
Imperfections to Allow」が０、「Minimum % of Matching Words」が100%の設定となっていて、完全一致のみをカウントしているせいであり、これらの値を変えて曖昧な一致を許容した場合には、これらの値は異なってくる。

また、図２の表の上には設定した「Comparison Rules」が出力されている。概ねデフォルトの設定だが、「Fewest Matches to Report」のみ、検証用にどんな場合でも結果が出力される「１」に設定してある。

続いて、「Side-by-Side」というリンクで示される、両文書の一致箇所の比較画面を図３に示す。ただし、本来の「Side-by-Side」画面は、名前の通り、２つの文書が左右横並びで表示されるが、ここでは紙面の都合により上下に並べ替えた形で示す（以下同様）。

図３で、アンダーラインで示されている部分(実際の画面では赤い字となる）が一致していると判定された箇所であり、途中の太字の部分は一致していないと判定された部分である。ここは、先頭部分が「Yes or No」と「No or Yes」と語順を入れ替えてあり、末尾が「WCopyFind」と「WcopyFind」と「C」の大文字・小文字が異なっているため、一致していない。この間の「When checked this parameter causes」も両文書で共通だが、単語数が５で「Shortest Phrase to Match」の６より小さいため、一致と見なされていない。興味深いのは、「to limit the character」で始まる部分が、上の文書（File L）では先頭にあり、下の文書（File R）では末尾に位置しているのだが、これらは一致すると判定されていることである。つまり、WCopyFindでは文書内の位置にかかわらず、一定数（「Shortest Phrase to Match」以上の単語が連続して共通であれば、一致していると判定されるのである。

続いて、同じ２つの文書に対して、「Ignore Letter Case」を設定した場合の結果（「Side-by-Side」画面）を図４に示す。



File Comparison Report

Produced by WCopyfind.4.1.4 with These Settings:

Shortest Phrase to Match: 6
Fewest Matches to Report: 1
Ignore Punctuation: No
Ignore Outer Punctuation: No
Ignore Numbers: No
Ignore Letter Case: No
Skip Non-Words: No
Skip Long Words: No
Most Imperfections to Allow: 0
Minimum % of Matching Words: 100

Perfect Match	Overall Match	View Both Files	File L	File R
42 (82% L, 82% R)	42 (82%) L; 42 (82%) R	Side-by-Side	simple2.txt	simple.txt

WCopyfind.4.1.4 found 1 matching pairs of documents.

図２　WCopyFindの結果出力画面

図４より、大文字・小文字の区別をなくしたことにより、「WCopyFind」と「WcopyFind」が一致していると見なされたことがわかる。同時に、これらが同一視されたことにより、「When checked this parameter causes」の５語も、一致したフレーズの一部として勘定されている。

図５は、「Ignore Letter Case」は指定せず、代わりに「Shortest Phrase to Match」の値を５に減らしたときの「Side-by-Side」画面である。

図５では、大文字・小文字の区別をしているために、「WCopyFind」と「WcopyFind」が一致していないことになるが、「When checked this parameter causes」の部分は連続した５単語が一致しているため、一致フレーズになっている。

２．日本語文書への適用

日本語文書に適用する上での問題点

次に、WCopyFindで日本語の文書を処理する場合について述べる。

WCopyFindは多言語対応を謳っており、「Language」の選択肢の中には「Japanese」もある。文書形式もMicrosoft Word文書、HTML文書、テキスト文書、pdf文書を扱えることになっているが、論文やレポートの剽窃検出という用途を考えると、Word文書を対象とする場合が多いと考えられる。本稿では日本語文書としてWord文書形式（.docx）のファイルに限定して検証を行った。

※筆者が試したところ、日本語のテキストファイル（.txt）の場合、処理はできるものの、通常のShift-JISコードでは「Side-by-Side」など出力画面の表示が文字化けする。他の様々な文字コードを試したところ、文字化けを回避するにはBOM（Byte Order Mark）付きのUTF8を使う必要があるようだ。

Wordの文書ファイルであれば結果出力は文字化けしないが、日本語の文書ファイルをそのままの形でWCopyFindで処理しても期待したような結果は得られない。なぜなら、日本語の文では、欧文のように単語と単語の間にスペースを入れる「分かち書き」がされないため、WCopyFindは日本語の「単語」を認識できないのである。そのため、唯一改行記号だけが文の区切りと見なされ、改行記号で区切られた部分を「単語」として扱ってしまう。日本語の文章の場合、Microsoft Wordなどワープロセッサで作成された文に改行記号を入れるのは、一般に段落の末尾であるため、結果的にWCopyFindは日本語の１段落を、一つの巨大な「単語」として扱うことになり、図２の結果画面に「Match」として示される数は段落数となってしまうのである。このことは文書間の記述の一致を検出するには著しく不利である。なぜなら、「Shortest Phrase to Match」をデフォルトの6に設定した場合、６段落連続して一致しないかぎり一致と見なされないからである。つまり、文書ファイルの相当部分を完全に「丸写し」したものでもないかぎり検出されないことになるので

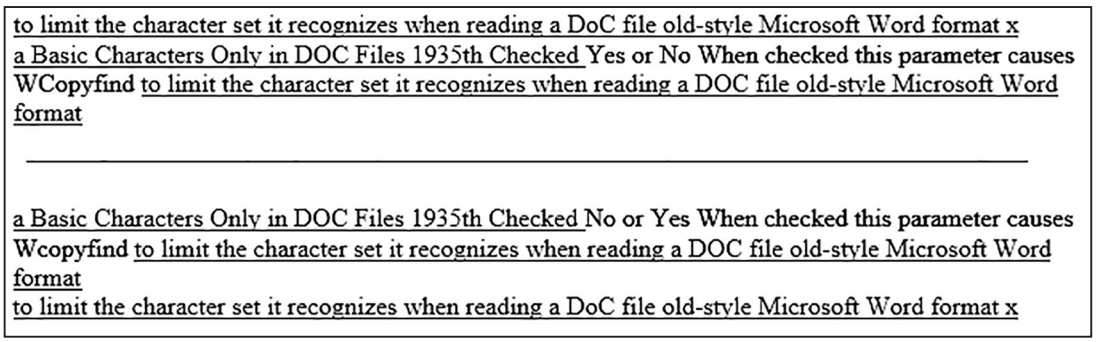


図３　Side-by-Sideの比較画面（上がFile L、下がFile R）

ある。「Shortest Phrase to Match」を最小の1に設定したとしても、一致の検出は1段落がまるまる同じである場合に限られることになる。レポートを丸写しするような場合であっても、一部の表現や語順を変えることで、違う文章に見せかけようとする「偽装工作」は容易に想像できるので、WCopyFindを単純に日本語文書に適用することはほとんど意味がない。

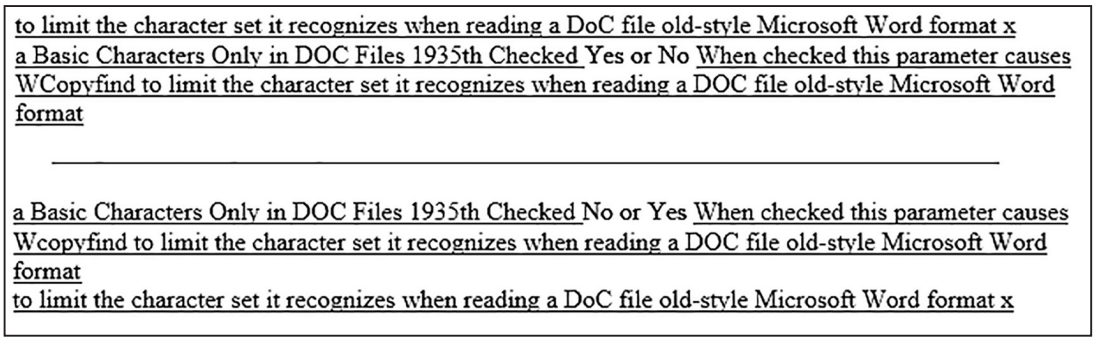
WCopyFindの基本設計である、「一定数以上の単語の連続した一致を見つける」という考えに合わせるとすれば、人の手作業で、あるいは日本語の構文解析ソフトウェアを利用することで、単語あるいは文節のレベルに分解して、各々の間にスペースを埋め込んだ文書を作成してからWCopyFindの処理にかけるといったことになる。しかし、そのようなことは現実的でない。そこで、簡易なやり方でこの点を代替する方法が必要である。

一つの解決策:文字単位への分解

筆者は、日本語文書内のすべての文字の間にスペースを埋め込むことでWCopyFindの認識する「単語」を「文字」に置き換えるやり方を試してみた。文章を単語や文節に分ける作業は手作業では非常に手間がかかるが、文字に分ける作業は簡単に機械的に処理できる。Microsoft Wordの場合、具体的には以下のように「置換」機能を使えばよい。

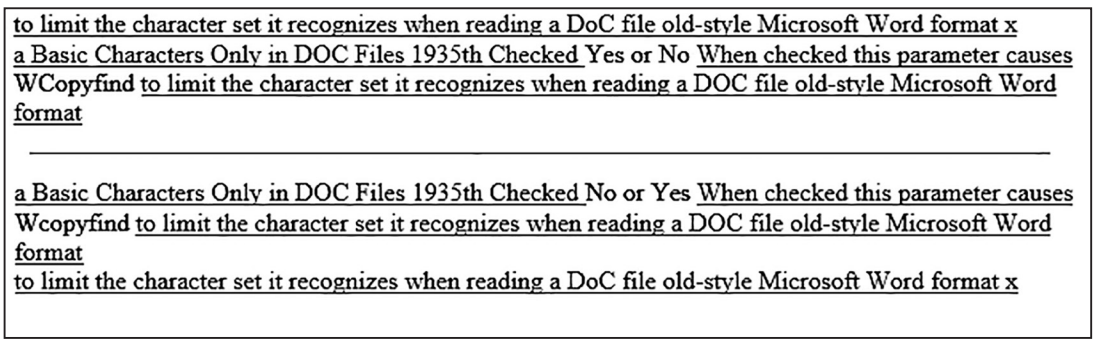
(1)「置換」画面を呼び出し、「オプション」を開いて「ワイルドカードを使用する」にチェックを入れる。
(2)「検索する文字列」に「(?)」を入れる。
(3)「置換後の文字列」に「\1 」を入れる（1の後ろは半角のスペース)。
(4)「すべて置換」を実行。

このような前処理を全部の文書に施した上でWCopyFindで処理するのである。すると、今度はWCopyFindが一つ一つの「文字」を「単語」


to limit the character set it recognizes when reading a DoC file old-style Microsoft Word format x
a Basic Characters Only in DOC Files 1935th Checked Yes or No When checked this parameter causes WCopyfind to limit the character set it recognizes when reading a DOC file old-style Microsoft Word format

a Basic Characters Only in DOC Files 1935th Checked No or Yes When checked this parameter causes Wcopyfind to limit the character set it recognizes when reading a DOC file old-style Microsoft Word format
to limit the character set it recognizes when reading a DoC file old-style Microsoft Word format x

図4　Side-by-Sideの比較画面（「Ignore Letter Case」設定後）


to limit the character set it recognizes when reading a DoC file old-style Microsoft Word format x
a Basic Characters Only in DOC Files 1935th Checked Yes or No When checked this parameter causes WCopyfind to limit the character set it recognizes when reading a DOC file old-style Microsoft Word format

a Basic Characters Only in DOC Files 1935th Checked No or Yes When checked this parameter causes Wcopyfind to limit the character set it recognizes when reading a DOC file old-style Microsoft Word format
to limit the character set it recognizes when reading a DoC file old-style Microsoft Word format x

図5　Side-by-Sideの比較画面（「Shortest Phrase to Match」を5とした場合）

とみなして処理を行うことになるため､段落を｢単語｣とみなすより細かい単位での処理になる。

この方法をとった場合､｢Shortest Phrase to Match｣の値はデフォルトの６ではやや短いかもしれない。｢連続した６文字が一致する｣という条件では、特定の言い回しや慣用表現などがある場合に､それだけで一致という判定をしてしまい、必要以上に一致を検出してしまう可能性があるからである。この値の適切さを一概に決めることは困難だが、15程度、つまり連続15文字の一致を検出するくらいに設定する必要があると思われる。同一のテーマで書かれた実際のレポート（部分的な丸写しなどの剽窃がないと考えられるもの）を使って筆者が試したところでは、この値を６のままにしておくと、一致があったという判定がしばしば見られるが、15程度に設定すると、こうした｢偶然の一致｣はほとんど見られなくなる。

もっとも、｢Shortest Phrase to Match｣の値が６のままであったとしても、剽窃の行われた文書の間では、他の文書間よりも一致の数や割合がずっと大きいだろう。また、この値を大きく設定した時に一致が見られたとしても、直ちに剽窃と判断するわけでもない。実際の判断は実際の文章を精読して行わなければならないことは言うまでもない。つまり、こうした一致の数や割合はあくまでも剽窃を疑うきっかけに過ぎないのである。そのためのヒントを与えてくれるツールとしてWCopyFindを捉えておくことを確認しておきたい。

引用文献

Bloomfield, Lou 2011 Software to Detect Plagiarism
< http://plagiarism.bloomfieldmedia.com/z-wordpress/software/>（January 16, 2016）