

Systems and Methods for Detecting Similarity of Documents

FIELD OF THE INVENTION

【0001】 This invention refers to a system and a method for detecting similarity of documents, in particular, to a system and a method for detecting similarity of both English and Chinese documents instead of mainly focusing in English documents as in prior arts. This invention involves Chinese word segmentation procedure using a machine learning based word tagging method, which is also advancement when compared to the prior arts.

BACKGROUND OF THE INVENTION

【0002】 Nowadays, it is increasingly convenient for people to acquire data via Internet, which incurs a side-function: plagiarism. Judgment of the plagiarism will take the following factors into account: 1) whether there are any contents (suspected contents) in one document similar to others; and/or 2) whether any references are given for the suspected contents.

【0003】 In view of the above, prevention of plagiarism becomes a hot topic in the art. There are a lot of papers dealing with this issue, for example,

- P. Clough. "Plagiarism in natural and programming languages: An overview of current tools and technologies". Research Memoranda: CS-00-05, Department of Computer Science, University of Sheffield, 2000;
- N.Shivakumar, H. "Garcia-Molina. SCAM: A copy detection mechanism for digital documents". Proceedings of the 2nd International Conference on Theory and Practice of Digital Libraries, Austin, Texas, 1995; and
- S. Brin, J. Davis, and H. Garcia-Molina. "Copy detection mechanisms for digital documents". In Proceedings of the ACM SIGMOD Conference, pages 398-409, 1995.

【0004】 In brief, known methods for detecting similarity of documents generally comprise steps of:

- normalizing each document in a document collection and segmenting each document into a plurality of contiguous chunks of tokens;
- selecting a certain number of representative chunks of tokens from the plurality of contiguous chunks of tokens;
- transforming the selected chunks of tokens into fingerprints and forming a signature from the fingerprints; and
- determining the similarity of two or more documents according to similarity of fingerprints of the documents. If identical fingerprints in two documents of the document collection are identified, the two documents will be regarded as similar. In other words, the similarity of two documents is dependent on the number of the identical fingerprints.

【0005】 However, the known methods are used for western languages (such as English) rather than eastern languages (such as Chinese), since certain factors considered in the methods, such as the relationship of capital letters and small letters, do not exist in Chinese at all. Moreover, word segmentation for Chinese languages is different from that for English languages. Thus, a system and a method for detecting similarity of documents in eastern languages are desired.

SUMMARY OF THE INVENTION

【0006】 The present invention is directed to provide a system and a method for detecting similarity of documents.

【0007】 According to one aspect of the present invention, a method for detecting similarity of documents is provided. The method comprises:

extracting a text from a first document and at least one second documents, respectively;

segmenting the text of the first document into a first plurality of units and the text of the at least one second documents into a second plurality of units; and

comparing the first plurality of segmented units with the second plurality of segmented units to determine whether the first document is similar to at least one of the at least one second documents.

【0008】 According to another aspect of the present invention, a system for detecting documents similarity is provided. The system comprises:

a text extraction module configured to extract a text from a first document and at least one second documents, respectively;

a segmentation module configured to segment the text of the first document into a first plurality of units and segment the text of the at least one second documents into a second plurality of units;

a comparison module configured to compare the first plurality of segmented units with the second plurality of segmented units to determine whether the first document is at least similar to at least one of the at least one second documents.

【0009】 According to the present invention, it is able to not only detect documents similarity in western language, but also optimize the document similarity detection in Chinese.

BRIEF DESCRIPTION OF THE DRAWINGS

【0010】 Figure 1 is a block diagram showing a system for detecting similarity of documents according to an embodiment of the present invention;

Figure 2 is a block diagram of a text processing module in Figure 1;

Figure 3 is a flow chart for training of steps in the LMR-RC tagging according to the present invention;

Figure 4 is a flow chart for tagging of steps in the LMR-RC tagging according to the present invention;

Figure 5 is an illustrative example of taking sentence representation codes as database indexes according to the present invention;

Figure 6 is an illustrative example of taking words as database indexes according to the present invention;

Figure 7 is an example of direct comparison for English sentences according to the present invention; and

Figure 8 is an example of partial comparison for Chinese sentences according to the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

【0011】 Detection of the similarity of documents according to the present invention is realized based on three basic assumptions.

【0012】 Firstly, the detection of similarity is carried out in view of text only, without checking non-text contents.

【0013】 Secondly, a sentence is deemed as components of the text to be detected. The larger the number of similar sentences is, the higher the similarity is.

【0014】 Thirdly, the similarity refers to the similarity in “impression”, rather than in literal expression. Examples which are similar in impression but different in literal expression include, but not limited to, singular nouns and plural nouns; verbs and changed verbs in tenses, uppercase and lowercase in English, traditional words and simplified ones in Chinese, and different computer character encodings, such as BIG5 and GB.

【0015】 Hereinafter, a system 1000 for detecting similarity of documents will be described in detail with reference to Fig.1.

【0016】 As shown in Fig. 1, the system 1000 comprises a conversion module 101, a text extraction module 102, a text processing module 103, a text encoding module 104, a comparison module 105, a report generation module 106 and a database 107.

【0017】 The conversion module 101 is configured to convert a document provided by a user into a computer-readable format. The conversion module 101 may be one of known devices like scanners, optical text recognizers, web interfaces, etc.

【0018】 The text extraction module 102 receives the document in computer-readable format from the conversion module 101, extracts text from the received document and transfers the extracted text to the text processing module 103.

【0019】 The text processing module 103 comprises a segmentation

sub-module 1031, a normalization sub-module 1032 and a high-frequency word removing sub-module 1033.

【0020】 The segmentation sub-module 1031 segments the extracted text into one or more units. A unit may be a paragraph, a sentence and/or a word. Namely, the extracted text may be segmented into one or more paragraphs, sentences and/or words or their combinations.

【0021】 When the text is segmented into paragraphs, a paragraph should be identified at first. Typically, a paragraph is identified according to normal paragraph identifiers like pressing a button “ENTER” two times, pressing a button “ENTER” and pressing a button “.”, etc.

【0022】 When the text is segmented into sentences, a sentence is identified according to a sign of “full stop”, a question mark, or the like.

【0023】 Segmenting the text into words needs to identify words first. In English, words are separated automatically by a separator like a spacer or a punctuation.

【0024】 Unlike English, words in one Chinese sentence are not automatically segmented. Thus, in order to segment Chinese text, the present invention proposes a Chinese segmentation technique based on Machine Learning methods, called “LMR-RC tagging technique”, including two stages, i.e., Regular-stage (R-stage) and Correctional-stage (C-stage).

【0025】 Specifically, according to the LMR-RC tagging technique, when a sentence in Chinese is to be segmented, first of all, each character in the sentence is assigned a label as a representative of the position at which the character is the most likely to be located in a word. The label may be “L”, “M”, “R” or “S”, wherein “L”, “M” and “R” indicate that the character locates at the beginning, at the middle and at the end of one word, respectively, while “S” indicates that the character itself constitutes one word.

【0026】 An example for assigning labels for a sentence “大衛喜歡吃揚州炒

飯” is shown in Table 1, wherein the sentence includes four words: 大衛, 喜歡, 吃, and 揚州炒飯. As known to the skilled in the art, each character is labeled according to the context of the character, namely, the “feature” of the character.

Word	大	衛	喜	歡	吃	揚	州	炒	飯
Label	L	R	L	R	S	L	M	M	R

Table 1

【0027】 Hereafter, the description is given on how the segmentation sub-module 1031 uses so-called “LMR-RC tagging technique” to realize the segmentation of a text with reference of Figs. 3 and 4. As mentioned above, LMR-RC tagging technique is carried out based on Machine Learning methods. LMR-RC includes a training step and a tagging step.

【0028】 Fig. 3 is a flowchart of step for training according to LMR-RC. As shown in Fig. 3, a training corpus 200 is input module 103 and divided into a first sub-corpus 200a and a second sub-corpus 200b (step S201), wherein the first sub-corpus 200a and the second sub-corpus 200b are not overlapping. The first sub-corpus 200a is used for training in the R-stage, while the second sub-corpus 200b is used for training in the C-stage. Preferably, the first sub-corpus 200a takes up about 70% of the whole corpus 200 and the corpus 200b takes up the rest 30%.

【0029】 At step S202, the feature of each character in the first sub-corpus 200a and the second sub-corpus 200b are extracted according to a R-stage feature template 203a to form a first feature collection 204a and a second feature collection 204b, wherein the first feature collection 204a includes all features extracted from the first sub-corpus 200a, and the second feature collection 204b includes all features extracted from the second sub-corpus 200b.

【0030】 Taking a sentence “32 個蘋果” as an example, the feature template 203a used for R-stage is defined as shown in Table 2, in which the word “個” is set as the current word.

	Feature Description	Extracted Feature
1	the current word, two preceding words of the current word, and two following words of the current word	$C_2 = "3"$ $C_{-1} = "2"$ $C_0 = "個"$ $C_1 = "蘋"$ $C_2 = "果"$
2	Each of two sequential words in the sentence	$C_2 C_{-1} = "32"$ $C_{-1} C_0 = "2 個"$ $C_0 C_1 = "個 蘋"$ $C_1 C_2 = "蘋果"$
3	one preceding word of the current word, and one following word of the current word	$C_{-1} = "2"$ $C_1 = "蘋"$
4	Whether the current character is a punctuation or not?	not
5	Whether any non-Chinese word exists in the two preceding words and in the two following words	A_2 and A_{-1} (because "3" and "2" are not Chinese words)
6	Whether the current word belongs to the same category as that of one preceding word or one following character	D_{-1} (because "3" and "個" belong to different categories)

Table 2

【0031】 Then, step S203 performs Machine Learning methods to learn the first feature collection 204a so as to generate a temporary tagging model, namely an R-stage model 206a. Any known machine learning algorithm can be used to generate the R-stage model 206a. In this invention, a learning algorithm called "Maximum Entropy Model" is preferable.

【0032】 The R-stage model 206a is a probability model and used for tagging the second sub-corpus 200b by taking the second feature collection 204b into account (step S204). That is, the R-stage model 206a is used for calculating a maximum-likelihood label of each character in the second sub-corpus 200b, so as to generate a R-stage tagging result 208.

【0033】 At step S205, features of the second sub-corpus 200b are extracted according to an R-stage/C-stage feature template 203b, and features of R-stage tagging result 208 are extracted, too, according to the same template 203b, so that a final feature collection 204c is formed. The R-stage/C-stage template 203b includes a R-stage template as shown in Table 2 and a C-stage template as shown in Table 3 given below.

【0034】 Step S206 performs Machine Learning methods for C-stage in view of both the final feature collection 204c and the first feature collection 204a to generate a tagging model 206b. The C-stage may use the same learning algorithm as that used in the R-stage or different one. Since both the first feature collections 204a and the final feature collection 204c are used in step S206, the tagging model 206b can be used for tagging in both R-stage and C-stage.

【0035】 Taking the sentence “32 個蘋果” as an example again, in which the R-stage tagging result 208 is “SSLMR”, the R-stage/C-stage feature template 203b is shown in Table 3, in which the word “個” is the current word.

	Feature Description	Extracted Feature
7	tagging results of the current word, two preceding words and two following words	$T_{-2} = "S"$ $T_{-1} = "S"$ $T_0 = "L"$ $T_1 = "M"$ $T_2 = "R"$
8	tagging results of each two sequential words	$T_{-2} T_{-1} = "SS"$ $T_{-1} T_0 = "SL"$ $T_0 T_1 = "LM"$ $T_1 T_2 = "MR"$
9	tagging results of one preceding word and one following word	$T_{-1} T_1 = "SM"$

Table 3

【0036】 Fig. 4 is a flow chart of step for tagging according to LMR-RC. At step S301, features of the received text 300 are extracted by using the R-stage feature

template 203a, so as to generate an R-stage feature collection 301a. Then, at step S302, the features in the R-stage feature collection 301a are tagged by using the model 206b, namely, the R-stage tagging, so as to generate an R-stage tagging result 302a. At step S303, features of the text 300 and features of the R-stage tagging result 302a are extracted according to the R-stage/C-stage feature template 203b, so as to form a final feature collection 301b. At step S304, the feature of the final feature collection 301b are tagged by using the R-stage/C-stage feature template 206b to generate a C-stage tagging result 302b. The tagging result 302b is a group of labeled Chinese characters. At step S305, valid label-sequences such as “LR”, “LMMMR” and “S” are identified from the tagging result 302b by using a pattern recognition technique like regular expression and the text 300 is finally segmented into word units according to the identified label sequences.

【0037】 It can be seen that, when the R-stage is completed, the result obtained from the R-stage is used for learning in the C-stage so as to make the C-stage tagging result precise. For example, the R-stage model 206a is generated by learning the first sub-corpus 200a. Since the accuracy of the learning and the information contained in the first sub-corpus 200a is limited, errors may occur. Taking “32 個蘋果” as an example, the correct tagging result should be “LRSLR”, but it is likely to generate a wrong tagging result “SSLMR” if only proceeding the R-stage tagging by using the R-stage model 206a. If a further tagging on this sentence by using the final tagging model 206b generated from the learning in the C-stage may result in a correct tagging result “LRSLR”.

【0038】 Return to Fig. 2. The above illustrates how the segmenting sub-module 1031 segments a text. After the text is segmented into units, the normalization sub-module 1032 receives the petitioned units from the segmentation sub-module 1031 and normalizes the segmented units so as to improve the efficiency of the detection of similarity. For example, as for an English text, the normalization may include a conversion from uppercase to lowercase, reversion from a compound word to its root or the like. For a Chinese text, the normalization may include a conversion between a traditional word and a simplified word and unifying the encoding of words. However, those skilled in the art can be understandable that the normalization is not limited to the above.

【0039】 For example, according to the ASCII codes, an interval between an uppercase and its corresponding lowercase is fixed. Thus, as long as an uppercase letter is identified, its corresponding lowercase is available by subtracting the length of the fixed interval from the position of the uppercase.

【0040】 It is preferable to convert all English words to their respective roots in order to improve the accuracy of the detection, including, but not limited to, reverting plural nouns into singular nouns, reverting various tenses of verbs into the present tense, etc. When processing a regular word, a known “Porter Stemming” algorithm is useful. Since most variations of a regular word occur at the ending of the regular word, the “Porter Stemming” algorithm modifies the ending of words based on a certain rule so as to realize the normalization. As to those irregular words, there are big differences between an irregular word and its variations and the variations do not follow any rule. Thus, when processing an irregular word, the present invention prefers to use a dictionary contrast list. That is, a contrast list is generated in advance to collect roots and their variations of all the irregular words. When one irregular word needs to be reverted, the irregular word is used as an index to retrieve its root from the list.

【0041】 Conversion between traditional texts and simplified texts is necessary when processing Chinese text normally. After completing the above conversion, the Chinese texts should be encoded by using a certain encoding rule, like Unicode.

【0042】 The high-frequency word removing sub-module 1033 is configured to remove words which often occur but do not have any substantial meaning, such as “a”, “an” and “to” in English and “的”, “是” and “和” in Chinese. These words are called “high-frequency words” or “stop words”. To remove the words, a high-frequency word list is generated. When a current word is processed, the high-frequency word removing sub-module 1033 scans the high-frequency-word list to determine whether the current word belongs to the list. If yes, the current word is regarded as a high-frequency word and removed.

【0043】 Return to Fig. 1. The output of the module 103 is received by the module 104, wherein the outputs may be paragraphs, sentences, words. The module

104 encodes the paragraphs, sentences and words to generate codes with a fixed length as a representative of each of the outputs and stores the codes into a database 107. In an embodiment of the present invention, the codes may be generated through one of the following algorithms MD5 algorithm, SHA-1 algorithm, SHA-2 algorithm, variations of these algorithms and other similar algorithms. The length of codes will be varied from the algorithm. For example, the length of the codes generated through MD5 algorithm is 128 bits, 160 bits of the length through SHA-1 algorithm, and 256 bits length of through SHA-2 algorithm.

【0044】 Since the codes have a fixed length and are able to represent those sentences with various lengths, the codes can be as an index for searching. As shown in Fig. 5, three kinds of codes are used as indexes for three sentences. Alternatively, words may be used as indexes, too, to retrieve corresponding sentences directly from the database 107, as shown in Fig. 6, in which eight different words are used as indexes for three different sentences.

【0045】 A document input into the present system 1000 will be extracted (module 102), text-processed (module 103), and encoded (module 104) as stated above, and then be stored in a database 107. Then, the document will be compared with those documents pre-stored in the database 107, so as to detect the similarity between the newly input document and those pre-stored ones.

【0046】 For example, when a document collection including two papers is input into the system 1000, each of the two papers will undertake the above-mentioned extracting, text-processing and encoding and then are stored in the database 107 so as to compare the two papers each other and compare each of the papers with all documents pre-stored in the database.

【0047】 Specifically, the module 105 carries out the comparison by taking a sentence as a basic unit (assumption 2). The comparison module 105 calculates the similarity by using a “direct comparison” or a “partial comparison”.

【0048】 In the direct comparison, if sentence A and sentence B are identical to each other after being processed by the text processing module 103, both sentences A and B are considered to be “similar”. Herein, the “identical” means that the lengths of

two sentences, words included in the two sentences as well as the order of the words are identical. Due to the identity, the codes of sentences A and B generated by the text encoding module 104 are identical, too. Thus, when the comparison module 105 searches the database by using the codes as index, the determination of the identity between sentences A and B will be done rapidly.

【0049】 An example of the direct comparison for two English sentences is shown in Fig. 7. As shown in Fig. 7, after being processed by the text processing module 103, both sentences “Peter is playing football” and “Peter plays football” are converted into “peter play football”. According to the algorithm of the direct comparison, sentences A and B are considered to be “similar”.

【0050】 In the partial comparison, if sentences A and B after being processed by the text processing module 103 comprise some words in common and the proportion between the common words and the total words in one sentence exceeds a predetermined threshold, sentences A and B are considered to be “similar”. The similarity of the sentences A to the sentence B is calculated by a rule of

$$\text{Similarity} = \frac{\text{number of common words in two sentences}}{\text{total words number in one sentence}} * 100\%$$

Specifically, the comparison module 105 calculates the number of common words in sentences A and B at first. As stated above, words can be used as an index. Therefore, the comparison may be carried out by searching words in the database and calculate common words number in sentence A so as to determine the similarity between two sentences A and B. An example of partial comparison is shown in Fig. 8. Two sentence A “由於天雨路滑，車輛的行駛速度都減慢了” and sentence B “因為天雨路滑的關係，車子行駛速度都減慢了” have similar meanings. After segmenting of sentences and removal of high-frequency words, 4 words in the sentence A and 5 words in sentence B are generated, respectively, wherein 3 words are in common. As a result, the similarity between sentence A and sentence B is $3/4=0.75$ (75%). If the predetermined threshold is 70%, sentence A is considered to be similar to sentence B.

【0051】 In addition, the similarity is accounted based on the total number of “similar” sentences. A total rate and an individual rate algorithm are recommended.

【0052】 The total rate algorithm is a measurement of the similarity of one document against a document collection as a whole. The collection includes documents provided by users, documents pre-stored in the database, or documents compared previously. The total rate is calculated as follow:

$$\text{Total rate of a document to a document collection} = \frac{\text{the number of similar sentences shared by the document and the document collection}}{\text{the numbers of sentences in the document}} * 100\%$$

【0053】 The individual rate algorithm is a measurement of the similarity of one document against another particular document and may be calculated as follow:

$$\text{Individual rate of a document A similar to a document B} = \frac{\text{the number of similar sentences shared by the document A and the document B}}{\text{the numbers of sentences in the document A}} * 100\%$$

When the similarity is determined by the document comparison module 105, the plagiarism probability may be determined accordingly. For example, the higher of the total rate of document A against a document collection is, the higher the plagiarism probability of document A is, because the higher total rate means that there are a large number of sentences in document A similar to those in the collection.

【0054】 While the modules 101-107 of Fig. 1 are described as seven separate elements, respectively, the invention is not limited thereto. It should be understood that those components may be implemented as lesser or more components as required. Similarly, while the module is described as a member comprising of three parts 1031-1033, the invention is not limited thereto. In addition, each of the modules in the invention may comprise one or more means for the respective function operation, as required. For example, the part 1031 may comprise one means for exacting the first feature collection 204a and the second feature collection 204b, or comprise two separate means for exacting the first feature collection 204a and the second feature collection 204b, respectively.

【0055】 The invention has been described in an illustrative manner. Obviously, many modifications, variations and equivalents of the present invention are possible

in light of the above teachings. It should be understood that the invention may be implemented by other systems and methods than those disclosed herein within the scope of the appended claims.

Claims

1. A method for detecting a similarity of a first document with at least one second documents, comprising:

5 extracting plain texts from the first document and the at least one second documents, respectively;

 tagging the extracted texts based on a tagging model;

 segmenting the extracted texts based on the tagged results from the first document into a first plurality of units, and the extracted texts based on the tagged results from the
10 at least one second documents into a second plurality of units, respectively; and

 comparing the first plurality of segmented units with the second plurality of segmented units to determine whether the first document is similar to at least one of the at least one second documents.

15 2. The method of claim 1, further comprising a step of normalizing the segmented units.

 3. The method of claim 2, further comprising a step of removing words that are frequently presented but do not have substantial meanings from the extracted text.

20

 4. The method of claim 1, further comprising:

 encoding each of the first and second plurality of units into one or more codes, each of which has fixed length; and

 storing each of the codes in a database as an index for the respective document.

25

 5. The method of claim 1, wherein the segmenting further comprises:

 extracting a first feature collection from the text based on a first feature template;

 tagging the first feature collection to generate a first tagging result based on the tagging model;

30 extracting a second feature collection from the texts and the first tagging result based on a second feature template;

tagging the second feature collection to generate a second tagging result based on the tagging model; and

segmenting the texts based on the second tagging results to generate the first and second plurality of units.

5

6. The method of claim 5, further comprising a step of generating the tagging model comprising:

providing a training corpus;

10 dividing the corpus into a first portion of the corpus and a second portion of the corpus randomly;

generating a temporary tagging model from the first portion of the corpus;

tagging the second portion of the corpus based on the temporary tagging model to generate the tagging model.

15

7. The method of claim 6, wherein

generating the temporary tagging model further comprises:

extracting features of the first portion of the corpus based on the first feature template; and

20 learning the extracted features to generate the temporary tagging model; and

wherein tagging the second portion of the corpus based on the temporary tagging model to generate the tagging model further comprises:

tagging the second portion of the corpus based on the temporary tagging model to generate a tagged result;

25 extracting features of the tagged result based on the second feature template; and

learning the extracted features to generate the tagging model.

30 8. The method of claim 7, wherein the second feature template comprises information of the first feature template.

9. The method of claim 1, wherein the comparing further comprises:

determining that one sentence of the first document is similar to the one of the at least one second documents in the case that the text units in the two sentences are identical.

- 5 10. The method of claim 1, wherein the step of comparing comprises:
determining that one sentence of the first document is similar to the at least one of the at least one second documents in the case that a rate of the number of common text units in two sentences to total text units number in one sentence exceeds a predetermined threshold.

10

11. The method of claim 9 or 10, wherein the first document is determined to be similar to the at least one of the at least one second documents, when a rate of the number of the similar sentences contained in the first document to the total sentences number of the contained in the first document exceeds a predetermined threshold.

15

12. A system for detecting similarity of documents, comprising:
a text extraction module configured to extract plain texts from a first document and at least one second documents, respectively;
a segmentation module configured to tag the extracted texts based on a tagging
20 model, and to segment the extracted texts based on the tagged results from the first document into a first plurality of units and the extracted texts based on the tagged results from the at least one second documents into a second plurality of units, respectively;
a comparison module configured to compare the first plurality of segmented units
25 with the second plurality of segmented units to determine whether the first document is similar to at least one of the at least one second documents.

13. The system of claim 12, further comprising:
an encoding module configured to encode each of the first and second plurality of
30 units into one or more codes, each of which has fixed length.

14. The system of claim 12, further comprising:

a normalization module configured to normalize the segmented units to reduce differences of the text in expression.

15. The system of claim 12, further comprising:

5 a high-frequency word removing module configured to remove words that are frequently presented but do not have substantial meanings from the segmented units.

16. The system of claim 12, wherein the segmentation module further comprises:

10 means for extracting a first feature collection from the text based on a first feature template;

means for tagging the first feature collection to generate a first tagging result based on the tagging model;

means for extracting a second feature collection from the text and the first tagging result based on a second feature template;

15 means for tagging the second feature collection to generate a second tagging result based on the tagging model; and

means for segmenting the text based on the second tagging results to generate the first and second plurality of units.

20 17. The system of claim 16, further comprising:

means for generating the tagging model, which is configured to divide a corpus into a first portion of the corpus and a second portion of the corpus randomly; to generate a temporary tagging model from the first portion of the corpus; and to tag the second portion of the corpus based on the temporary tagging model so as to generating
25 the tagging model.

18. The system of claim 13, further comprising:

means for storing encoded codes based on sentences, phrases, and words in the first document and the at least one second documents.

30

19. The system of claim 18, wherein the means for storing is a database system and/or other data storage media.